jamk.fi

Langattoman verkkoratkaisun toteutus SDN-verkkoon

Timo Ihanainen

Opinnäytetyö
Syyskuu 2016
Tekniikan ja liikenteen ala
Tietoverkkotekniikan koulutusohjelma

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Ihanainen, Timo	Opinnäytetyö, AMK	13.12.2016
	Sivumäärä 80	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Langattoman verkkoratkaisun toteutus SDN-verkkoon

Tutkinto-ohjelma
Tietotekniikan koulutusohjelma

Työn ohjaaja(t)
Antti Häkkinen, Sampo Kotikoski

Toimeksiantaja(t)
Cyber Trust -projekti, Jyväskylän ammattikorkeakoulu, Tuure Valo

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi Jyväskylän ammattikorkeakoulun projektiryhmä, joka on osallisena suurempaa DIMECC:n rahoittamaa Cyber Trust-hanketta. Hankkeen tavoitteena on edistää suomalaista kyberturvallisuuden tasoa ja kannustaa organisaatioita yhteistyöhön. Ohjelma jakautuu useaan tutkimusalueeseen, josta JAMK keskittyy tietoturvaan uusissa verkkotekniikoissa.

Toimeksiantona opinnäytetyölle oli tutkia, kuinka saadaan kiinteään SDN-verkkoon liitettyä langaton lähiverkko niin, että se täyttää SDN-tekniikan ominaisuudet. Langattoman lähiverkon tukiasemat tuli lisäksi rakentaa käyttämällä Raspberry Pi-tietokoneita. Lisäksi vaatimuksena oli toteuttaa verkkoon liittyville asiakaslaitteille kevyt autentikaatio, jonka yhteydessä tuli selittää verkon käyttöehdot.

Työn ympäristö oli rakennettu sekä fyysisistä että virtuaalisista komponenteista. SDN-kontrolleri (ONOS) ja uuden aliverkon langattomalle lähiverkolle tekevä palomuuri (PfSense) oli työssä virtualisoitu, mutta topologia on suunniteltu työssä kuitenkin niin, että verkko voidaan rakentaa tuotantokäyttöön tukiasemien puolesta käyttämällä opinnäytetyön dokumentaatiota.

Työn lopputuloksena saatiin rakennettua langaton lähiverkko, joka saatiin liitettyä osaksi yksinkertaista kiinteää SDN-verkkoa. Tukiasemat onnistui tehdä Raspberry Pi-tietokoneista käyttämällä Open vSwitch-virtuaalikytkintä, sillä siinä löytyy tuki OpenFlow-protokollalle. Verkkoon liityttäessä kevyt autentikaatio ja käyttöehdot suoritettiin palomuurin Captive Portal -ominaisuudella.

Avainsanat (asiasanat)
SDN, WLAN, Raspberry Pi, Hostapd, OpenFlow, Open vSwitch

Muut tiedot

jamk.fi

Description

Author(s) Last name, First name Ihanainen, Timo	Type of publication Bachelor's thesis	Date 13.12.2016
		Language of publication: Finnish
	Number of pages 80	Permission for web publication: x

Title of publication

Implementing a wireless network solution for SDN network

Degree programme

Information Technology

Supervisor(s)

Antti Häkkinen, Sampo Kotikoski

Assigned by

Cyber Trust-project, Jyväskylä University of Applied Sciences, Tuure Valo

Abstract

The thesis was assigned by a working group of JAMK University of Applied Sciences, which is part of the Cyber Trust -scheme funded by DIMECC. Scheme's main goal is to improve the quality of Finnish cyber security and encourage a collaboration between organizations. The research program is shared to themes and JAMK is focused on information security in new network technologies.

The assignment was to research how to add a wireless local area network to be a part of the solid SDN network. Wireless access points had to be made by using Raspberry Pi-computers. In addition the network had to consist a lightweight authentication with a terms of use. User had to be able to accept the terms or the connection couldn't be successful.

The topology was partially virtualized. Access points and basic switching devices were physical. SDN-controller (ONOS) and firewall (PfSense) were virtual machines in the same physical computer. The firewall made a new subnet for wireless client devices. However the topology was premeditated that the network can be build for the use of main trusted network utilizing this document as a guide.

The wireless local area network was successfully attached to be a part of a solid SDN-network. Raspberry Pi-devices were able to configure as access points using Open vSwitch multilayer virtual switch because it includes a support for OpenFlow-protocol. A light authentication and network's terms of use were accomplished using a Captive Portal, which is a feature in the firewall.

Keywords/tags (subjectshttp://vesa.lib.helsinki.fi/)

SDN, WLAN, Raspberry Pi, Hostapd, OpenFlow, Open vSwitch

Miscellaneous

Sisältö

Kuviot

Taulukot

1 Lähtökohdat

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi JAMK eli Jyväskylän ammattikorkeakoulu. JAMK on vaikuttava ja kansainvälinen ammattikorkeakoulu, jossa opiskelijoita on yli 8500 ja henkilökuntaa noin 700. JAMK :n organisaatio koostuu viidestä yksiköstä: hallinto, ammatillinen opettajakorkeakoulu, liiketoimintayksikkö, hyvinvointiyksikkö ja teknologiayksikkö. Eri tutkintomahdollisuuksia löytyy seitsemältä eri alalta ja toimipisteitä on usealla kampuksella. JAMK on myös kansainvälisesti rikas ammattikorkeakoulu, sillä sinne saapuu vuosittain jopa satoja ulkomaalaisia vaihto- ja tutkinto-opiskelijoita. (Tietoa JAMKista n.d.)

Itse opinnäytetyön tilauksesta vastasi Cyber Trust -hankkeen projektiryhmä Jyväskylän ammattikorkeakoulusta. Cyber Trust -hanke on DIMECC:n neljävuotinen ohjelma (1.5.2015-30.4.2019), jonka tavoitteena on kehittää suomalaista kyberturvallisuuden tasoa erilaisissa verkkotekniikoissa ja -ympäristöissä sekä kannustaa yhteistyöhön eri yritysten ja organisaatioiden kanssa. Erityisesti keskitytään siis tietoturvallisiin palveluihin, alustoihin ja verkkoihin sekä uhkien tunnistamiseen. Kirjoitushetkellä (syksy 2016) hankkeessa oli mukana 25 yritystä ja 9 organisaatiota (DIMECC: Cyber Trust Program n.d).

Hankkeen tutkimusalueet on jaettu neljään teemaan, jotka on nimetty *Work Packageiksi* eli tuttavallisemmin työpaketeiksi. JAMK kuuluu työpakettiin 3, jonka virallinen nimi on Security in New Network Technologies. Työpaketissa nimensä mukaisesti perehdytään tietoturvan parantamiseen uudemmissa tietoverkkotekniikoissa ja -sovelluksissa, sillä yhä useammin verkot rakennetaan käyttämällä SDN-teknologiaa sekä verkkolaitteita ja verkkotoiminnallisuuksia voidaan virtualisoida (DIMECC: Securing platforms and networks n.d.).

1.2 Toimeksianto ja tavoitteet

Tehtävänä opinnäytetyössä oli suunnitella ja rakentaa langaton lähiverkko, joka on yhteydessä yksinkertaiseen SDN -hybridiverkkoon. Sana hybridiverkko tulee siitä, että

testiverkon rakentaminen voitiin tehdä sekä fyysisillä laitteilla että virtuaalisesti. Todellisuudessa itse SDN-tekniikalla hallittavan kytkinverkon hyöty tulisi esiin silloin, kun se on monipuolinen ja sisältää useita NFV-tekniikalla toteutettuja komponentteja (palomuurit, IDS/IPS-palvelut yms), mutta tässä työssä ”kiinteästä” SDN-verkosta tehtiin suhteellisen yksinkertainen, koska työn päämäärä oli tuoda siihen langaton verkkoratkaisu tietyillä vaatimuksilla.

WLAN -tukiasemat tuli rakentaa Raspberry Pi -piirilevytietokoneista ja konfiguroida ne käyttämällä hyväksi avoimen lähdekoodin käyttöjärjestelmää. Tavoitteena oli myös selvittää protokolla, jolla tukiasemat saadaan yhteyteen SDN-kontrollerin kanssa, jolloin niitä voidaan keskitetysti hallita ja monitoroida aivan kuten SDN-kytkinverkkoakin. Lisäksi verkon käyttämisen ehdot tuli jollain ratkaisulla ilmoittaa käyttäjille ennen kuin itse yhteys verkkoon muodostetaan.

Tavoitteena oli lisäksi esitellä teoriassa SDN -arkkitehtuuri, protokollia, joiden avulla verkkolaitteet saadaan keskustelemaan kontrollereiden kanssa ja tuotteet/ohjelmat, joilla tämä työ oli mahdollisuus toteuttaa aiemmin mainittujen vaatimusten puitteissa. Lisäksi tarkoituksena oli lyhyesti verrata SDN-arkkitehtuurin etuja ja haittoja fyysisillä laitteilla toteutettuun perinteiseen Ethernet-verkkoon.

2 Software Defined Networking

2.1 Yleistä

SDN eli *Software Defined Networking* on tekniikka, joka tarjoaa uuden lähestymistavan ohjata verkkoa. Pääideana on, että verkon ja sen laitteiden käyttäytymistä kontrolloidaan, muutetaan ja hallinnoidaan keskitetysti jakamalla arkkitehtuuri omiin tasoihin (Haleplidis, Pentikousis, Denazis & Hadi Salim 2015, 1).

Yksinkertaisimmillaan se voidaan kuvata seuraavasti: sovellustasolla (*Application plane)* luodaan äly kontrollitasolle (*Control Plane*). Kontrolleri on piste, josta tämä äly jaetaan verkon laitteille ja elementeille ja verkkoa hallitaan tietyn protokollan avulla yhdestä pisteestä. Verkkolaitteilla oleva välitystaso (*Data Plane)* vastaa liikenteen kytkemisestä ja reitittämisestä sen perusteella, millaiset ohjeet se on saanut ylemmältä kontrolleritasolta. Tällainen tekniikka mahdollistaa dynaamisesti hallittavan ja

kustannustehokkaan verkon, joka adaptoituu hyvin nykyajan verkkoliikenteen trendeihin (Open Networking Foundation 2014.)

2.2 Miksi?

Uutta tekniikkaa kuitenkaan ei ole luotu ilman sen tarvetta. Perinteisemmät tietoverkkoarkkitehtuurit eivät aina enää tänä päivänä vastaa yritysten ja loppukäyttäjien vaatimuksia, ja SDN on yksi tekniikka, jolla pyritään täyttämään heikot kohdat.

Räjähdysmäisesti kasvanut mobiililaitteiden määrä, palvelinvirtualisointi, konesalipalvelut ja pilvipalvelut haastavat perinteisen staattisen tietoverkon, jolloin perinteinen asiakas-palvelin -malli oli hallitseva. Nykyään käyttäjät ja sovellukset käyttävät useita tietokantoja ja palvelimia. Lisäksi pääsy näihin resursseihin on saatava millä laitteella, mistä tahansa ja milloin tahansa, mikä osaltaan muuttaa perinteisen tietoverkon toimintamallia (ONF: Software-Defined Networking. 2012 1-3.)

Yritysverkot ovat ottaneet nopeasti käyttöön julkiset ja sisäiset pilvipalvelut, mikä on johtanut niiden ennakoitua rajumpaan kasvuun. Tämä edellyttää sitä, että pääsy resursseihin on mahdollinen aina ja mistä tahansa tietoturvallisesti, resurssien tai datavaraston määrää voidaan kasvattaa tai vähentää nopeasti, organisaatioissa voi tulla rakenteellisia muutoksia ja konesalienkin massiiviset tietomäärät käyttävät suuren osan kaistanleveydestä. Tällaiset tilanteet johtavat siihen, että perinteinen ja manuaalisesti hallittava ja konfiguroitava tietoverkko ei skaalaudu nopeisiin muutoksiin tehokkaasti tai välttämättä lainkaan, vaan tarvitaan elastisesti skaalautuva hallinta tietoverkolle, jossa toimintoja on automatisoitu. (ONF: Software-Defined Networking. 2012, 3-4).

Yksi suuri rajoitus vanhoissa tietoverkkotekniikoissa on sen laaja kirjo erilaisia protokollia, jotka on luotu ratkaisemaan tietty ongelma, mahdollistamaan jokin palvelu tai parantamaan yhteyksiä, mutta ovat kuitenkin ikään kuin eristyksissä toisistaan. Pitkässä juoksussa tämä jatkuva protokollien kehitys tuo esiin kuitenkin selvän ongelman: monimutkaisuuden. Vanhassa tietoverkossa jo yhden aktiivilaitteen lisääminen järjestelmään voi vaatia usean verkkolaitteen uudelleenkonfigurointia käsin, VLAN-verkkojen ja ACL-listojen päivittämistä ja monien muiden protokollariippuvaisten me-

kanismien asetusten muuttamista. Lisäksi on mahdollista, että tapauksessa lisätty aktiivilaite ei sovi laitemallinsa tai -valmistajansa, ohjelmistoversionsa tai vaadittujen protokollien puutteidensa takia tietoverkkoon, joten staattisen tietoverkon monimutkaisuus on aito rajoite. (ONF: Software-Defined Networking 2012, 4-5.)

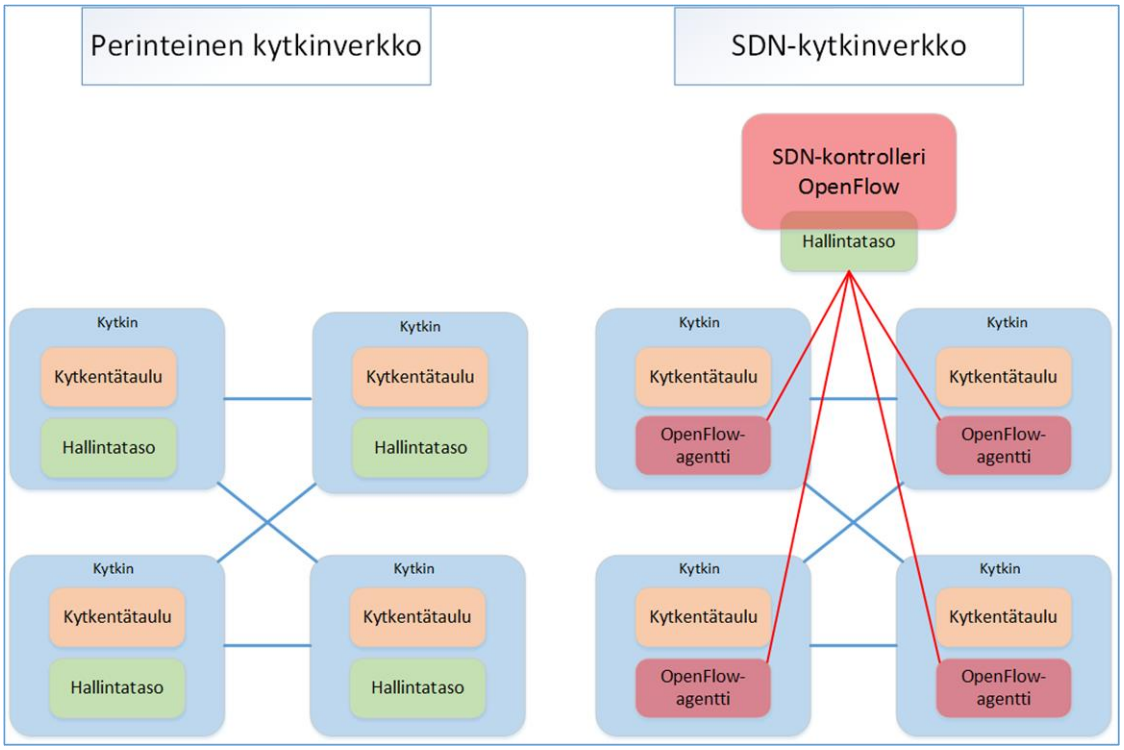
Todella suuret palveluntarjoajat (Google, Facebook jne) omistavat useita suuria konesaleja yhteydessä toisiinsa, joissa on todella suuri määrä virtuaalisia ja fyysisiä palvelimia, joiden täytyy olla valmiina skaalautumaan ja venymään tilanteissa, jossa esimerkiksi kuormantasausta koetellaan tai laskentatehoa vaaditaan äkillisesti lisää. On luonnollisesti selvää, että tällaisessa tilanteessa palvelinten manuaalinen konfiguraatio on mahdotonta.

Kaikkiin näihin rajoitteisiin on yhtenä vastauksena luotu Software-Defined Networking. SDN -verkko on vähemmän riippuvainen tietyistä valmistajista tai protokollista, verkkolaitteiden ja -elementtien hallinta ja konfigurointi on keskitettyä, palveluita ja toimintoja voidaan automatisoida. Ennen kaikkea ratkaisu tuo joustavuutta ja mahdollisuuden skaalautua muutoksiin ilman monimutkaisia ongelmia. (ONF: Software-Defined Networking 2012, 5-8.)

2.3 Arkkitehtuuri

2.3.1 Yleistä

SDN :n arkkitehtuurissa olennaisinta verrattuna perinteiseen tietoverkkoon on, että ennen verkkoelementit sisälsivät sekä kontrolli- että edelleenvälitystason samalla laitteella ja yhden rajapinnan takana. Tämä tarkoittaa sitä, että yhden rajapinnan lävitse kulkee usean eri protokollan liikennettä (reititysprotokollat, MAC -kyselyt, kytkentä- ja reitityspäätökset, hyötydata yms), joka hidastaa ja monimutkaistaa liikennettä. SDN:ssä tasot ovat eritelty toisistaan ja niillä on oma rooli sekä rajapinta. Tämä mahdollistaa keskitetyn hallinnan yhteen pisteeseen sekä yksinkertaistaa rajapinnan liikennettä. Siihen, miten moneen tasoon arkkitehtuuri jaetaan, on useita määritelmiä, mutta tässä työssä otetaan esille kaksi omaksutuinta määritelmää. Käytännössä idea on kuitenkin niissä identtinen, ja niiden ero vanhan kytkinverkon arkkitehtuuriin käy ilmi kuviossa 1.

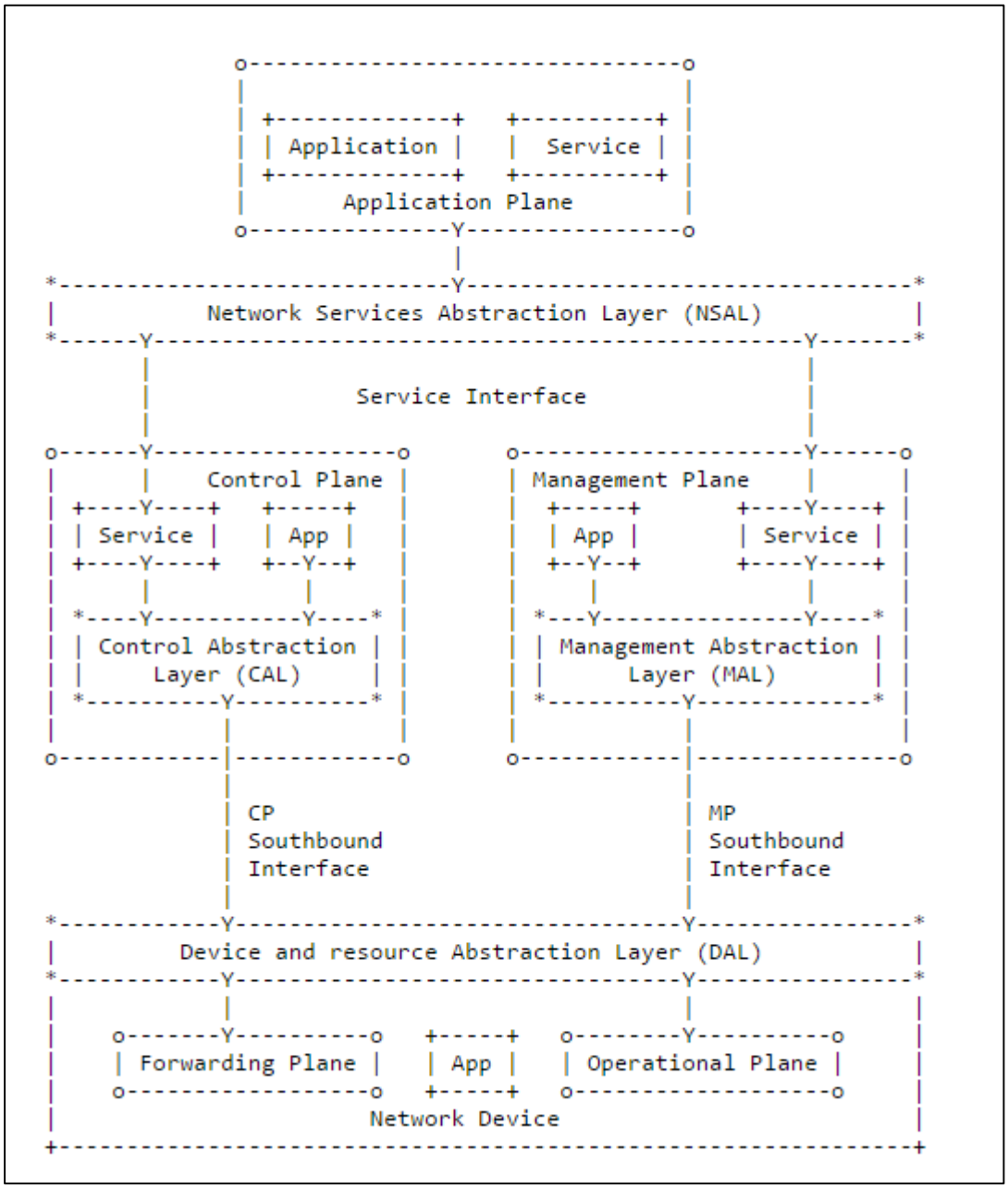


Kuvio 1. Perinteisen kytkinverkon ero SDN -kytkinverkkoon

2.3.2 RFC 7426 -standardin määritelmä

IETF-organisaatio, joka suuri kansainvälinen organisaatio, joka muodostuu tietoverkon suunnittelijoista, operaattoreista ja laitevalmistajista, jotka tutkivat Internet-arkkitehtuurin kehitystä. IETF julkaisee sivustollaan lukuisia erilaisia RFC -dokumentaatioita (*Request for comments)*, jotka voivat olla sisällöltään esim. virallisia Internet -standardeja, protokollia, käsitteitä, tietoja ohjelmista ja paljon muita tietoteknisiin menetelmiin liittyviä asiakirjoja (About the IETF n.d).

RFC 7426 on yksi näistä standardeista, ja sen sisältö on määritellä SDN :n arkkitehtuuri selittämällä eri tasot, tasojen väliset rajapinnat ja niiden väliset funktiot. Dokumentti on vastaus siihen, miten SDN-tekniikka tarkalleen toimii. Kuviossa 2 esitellään kuvan muodossa, miten tasot erotellaan keskenään.



Kuvio 2. RFC 7426 -standardin määritelmä SDN -arkkitehtuurista (Haleplidis ym. 2015)

RFC 7426:n määritelmästä nähdään (ks. kuvio 2), miten eri tasot on eroteltu toisistaan. Tasoja on viisi, jotka voidaan hierarkian myötä jakaa sen mukaan, mikä verkkoelementti niistä on vastuussa. Jaetaan verkkoelementit luokiksi verkkolaite, SDN -kontrolleri ja applikaatio.

Kuvan alalaidassa (kts. kuvio 2) olevat välitystaso (*Forwarding Plane)* ja toimintataso (*Operational Plane)* ovat verkkolaitteen vastuulla. Välitystason tehtävä on käsitellä paketteja sen kytkentäpolulla sen mukaan, millaiset ohjeet se saa ylemmältä tasolta.

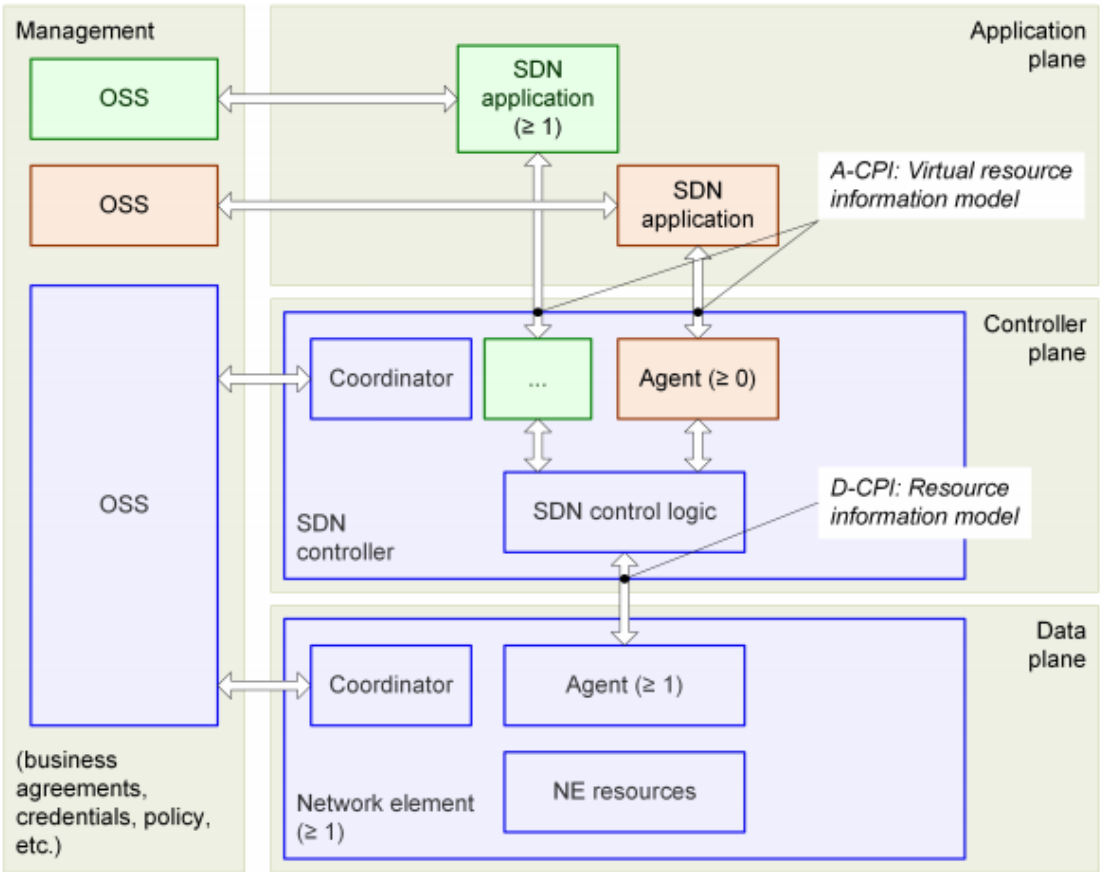
Välitystaso voi esim. edelleenvälittää, pudottaa tai muuttaa paketteja. Toimintatason tehtävä sen sijaan on ylläpitää tietoa verkkolaitteen toiminnallisesta tilasta, kuten vapaiden porttien määrä ja porttien statusta. Toimintataso ylläpitää myös tietoa itse laitteen resursseista esim. keskusmuistista tai prosessorin käyttöasteesta. Verkkolaitteen tasot ovat DAL-abstraktiotason kautta omilla rajapinnoillaan yhteydessä SDN -kontrolleriin. (Haleplidis ym. 2015, 9.)

Kuvion 2 keskellä esitetyt kontrollitaso (*Control Plane)* ja hallintataso (*Management Plane)* ovat SDN -kontrollerin vastuulla. Kontrollitason tehtävä on luoda päätöksiä, kuinka verkkolaitteen tulee välittää paketteja, joita se vastaanottaa ja jakaa nämä päätökset toteutukseen yhdelle tai useammalle verkkolaitteelle. Kontrollitaso vaikuttaa CPSI -rajapinnan (*CP Southbound Interface*) kautta verkkolaitteen välitystasoon. Kontrollitaso lisäksi hienosäätää kytkentätauluja, jotka sijaitsevat verkkolaitteen välitystasolla. Hallintataso sen sijaan vaikuttaa enemmän verkkolaitteen toimintatasoon MPSI -rajapinnan (*MP Southbound Interface)* kautta ja sen tehtävä on monitoroida, konfiguroida ja ylläpitää verkkolaitteita. (Haleplidis ym. 2015, 10-12.)

Sovellustaso (*Application plane)* kuuluu luokkaan applikaatio, johon SDN -kontrollerin tasot ovat yhteydessä NSAL -abstraktiotason kautta yhdellä rajapinnalla. Sovellustasolla tietyillä palveluilla ja sovelluksilla määritellään, miten tietoverkon halutaan käyttäytyvän. Sovellustasolla luodaan alkuperäinen äly, jonka mukaan SDN -kontrolleri tekee päätöksenä suhteen. (Haleplidis ym. 2015, 10-12.)

2.3.3 ONF-organisaation määritelmä

ONF (*Open Networking Foundation)* on voittoa tavoittelematon ja käyttäjälähtöinen organisaatio, joka on omistautunut kehittämään Software-Defined Networking -tekniikan käyttöönottoa. Organisaatioon kuuluu laajalla rintamalla jäseniä aina pienistä startup -yrityksistä suuriin verkko-operaattoreihin ja palveluntarjoajiin. ONF keskittyy kehittämään sovelluksia, standardeja ja protokollia SDN:ään, josta merkittävin toistaiseksi on *OpenFlow* -protokolla. Organisaation määritelmä SDN -arkkitehtuurista käy ilmi kuviosta 3. (ONF: What is ONF? n.d.)



Kuvio 3. ONF:n määritelmä SDN -arkkitehtuurista (Open Networking Foundation, 2014)

Verrattuna RFC 7426 -standardiin idea on pääosin täysin sama, joskin suurimpana poikkeuksena on RFC -standardissa olevien välitys- ja toimintatasojen yhdistäminen yhdeksi tasoksi (*Data plane)* ja käsittää niin ikään verkkolaitteen. Tämän tason sisälläkin on kuitenkin nähtävissä, miten rajapinnat on eroteltu erikseen hallinta- (*Management)* ja kontrollitasolle (*Controller plane*).

ONF:n määritelmästä löytyy kuitenkin joitakin yksityiskohtia, joita ei RFC-standardissa ole. Kuviossa 3 näkyvien agenttien rooli on tukea perustana olevien resurssien jakamista esimerkiksi, että mitkä elementin portit ovat SDN-kontrolloituja, tai tiedottaa yksityiskohtia virtuaalisesta verkosta, jos verkossa on useamman eri asiakkaan palveluita, jotka tulee erotella toisistaan (Open Networking Foundation 2014, 6.)

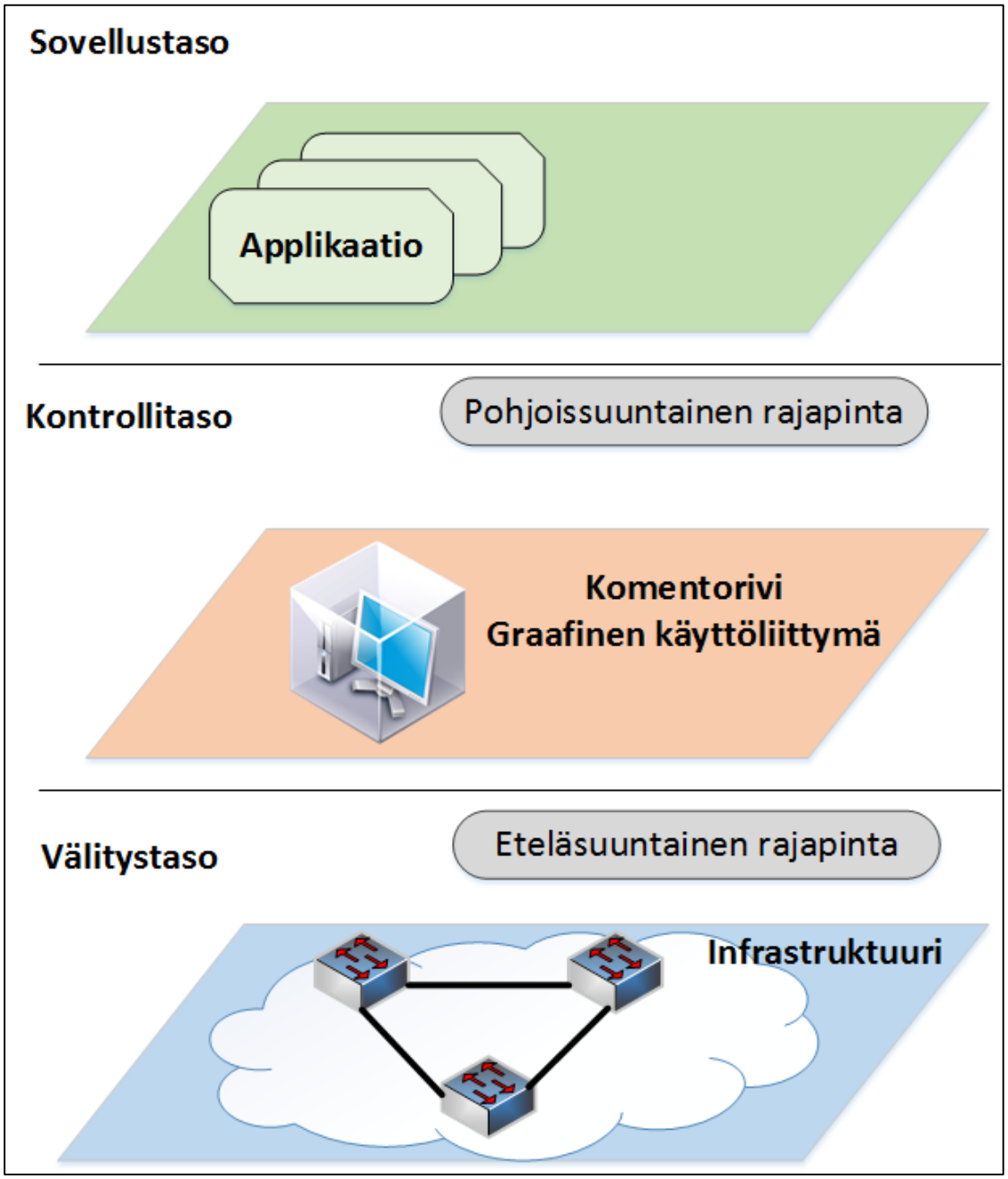
Lisäksi kuviossa pyritään väreillä painottamaan luottamukselliset toimialueet. Sininen kuvastaa koko tietoverkon palveluntarjoajaa, kun taas värit punainen ja vihreä ilmaisevat asiakkaita sekä erillään olevia kokonaisuuksia palveluntarjoajan luottamuksellisen toimialueen sisällä. (Open Networking Foundation 2014, 7.)

2.4 SDN-kontrolleri

2.4.1 Yleistä

SDN -kontrollerit ovat käytännössä tietoverkon ”aivot”. Se voi olla joko virtuaalisesti toteutettu tai fyysinen palvelin, jossa on asennettuna jokin SDN -ympäristöihin soveltuva käyttöjärjestelmä. Näitä käyttöjärjestelmiä löytyy myös useina avoimen lähdekoodin ratkaisuina, jotka ovat pääosin Linux -pohjaisia. SDN -kontrollerialusta tyypillisesti koostuu kokoelmasta erikseen lisättävistä moduuleista, joilla voidaan suorittaa tietoverkossa eri tehtäviä. Laajennuksia ja moduuleita voidaan lisätä jälkeenpäin, jos nähdään tarpeelliseksi parantaa toimivuutta tai tukea kehittyneempiä ominaisuuksia(What are SDN controllers? n.d.)

Tämä strateginen keskitetty piste hallitsee välitys ja toimintatasojen verkkolaitteita ns. eteläsuuntaisen ohjelmointirajapinnan (*Southbound API)* kautta. Pohjoissuuntaisen ohjelmointirajapinnan (*Northbound API*) kautta järjestetään tietoverkon logiikka. Näiden rajapintojen looginen sijainti käy ilmi kuviosta 4 (What are SDN controllers? n.d.)



Kuvio 4. Ohjelmointirajapintojen sijainti SDN –verkossa

2.4.2 Topologian rakentaminen

Yleisimmät protokollat, joiden avulla kytkimet ja reitittimet keskustelevat kontrolle-reiden kanssa, ovat OpenFlow, OVSDB, YANG ja NetConf. Kytkinverkossa laite on siis yhteydessä kontrolleriin, jonka ohjeiden mukaan se tekee tehtäviään ja ilmoittaa kontrollerille tietoja mm. tilasta, porteista ja vuotaulusta. Kontrolleri ei kuitenkaan

saa rakennettua kuvaa koko verkon topologiasta, sillä se ei ymmärrä oletuksena kytkinten välisiä linkkejä. Tähän käytetään LLDP- (*Link Layer Discovery Protocol)* ja OFDP-protokollaa (*OpenFlow Discovery Protocol)*.

LLDP -protokolla sallii tietoverkon solmujen, tässä tapauksessa kytkinten, mainostaa muille LAN -verkon solmuille laitteensa tietoja, kuten mm. portit, nimi, VLANit, laitteen toiminnallisuudet, MAC-osoitteen jne. LLDP on tyypillisesti toteutettu Ethernet-kytkimillä, jossa ne lähettävät ja vastaanottavat LLDP-paketteja jokaisen portin kautta tietyllä Multicast -osoitteella. Nämä paketit kuitenkin kulkevat vain yhden hypyn päähän ja niitä ei välitetä eteenpäin (Pakzad, Portmann, Lum Tan & Indulska 2014.)

OpenFlow-kytkimet eivät tosin tue minkäänlaista yhteistä tapaa rakentaa tietoa topologiasta, joten se on kokonaan SDN-kontrollerin vastuulla. Kyseessä on mekanismi, josta käytetään termiä OFDP, joka ei tosin ole virallinen standardi. OFDP käyttää hyväksi LLDP -protokollaa, mutta toimii silti hieman eri tavalla, koska OF-kytkin ei sellaisenaan voi lähettää ja vastaanottaa LLDP -viestejä. Mekanismissa käytetään *Packet-out* ja *Packet-in* -viestejä. Packet-out -viestissä pakotetaan LLDP-paketit ulos verkon jokaisen kytkimen jokaisesta portista. Kytkimiin on erikseen määritetty sääntö vuotaulussa, jonka mukaan LLDP -paketit lähetetään aina kontrollerille Packet-in -viestinä huolimatta siitä, mistä portista paketti tulee. Prosessissa viestit toistetaan tietyin väliajoin jokaisella kytkimellä tietoverkossa, joten kontrollerilla pysyy jatkuvasti tieto kytkinten välisistä linkeistä ja näin ollen ajan tasalla oleva topologia. (Pakzad, Portmann, Lum Tan & Indulska 2014.)

2.4.3 Eteläsuuntainen rajapinta

Kuten kuviosta 4 todettiin, SDN -arkkitehtuurissa SDN -kontrolleri keskustelee verkkolaitteiden kanssa Southbound API -rajapinnan kautta. Rajapinta mahdollistaa kontrollerin tehdä dynaamisesti muutoksia verkon laitteille vaatimuksien ja tarpeiden vaatiessa. Protokolla, jonka avulla keskustelu kontrollerin ja verkkolaitteen välillä tapahtuu, voidaankin ikään kuin mieltää samalle ”tasolle” tämän API -rajapinnan kanssa, jolloin sen funktio on helpompi ymmärtää (What are SDN Southbound APIs? n.d.)

Vaikkei OpenFlow olekaan ainoa protokolla tähän tarkoitukseen, niin on se tunnetuin ja suosituin myös senkin takia, että kaikki suurimmat kytkin- ja reititinvalmistajat ovat ilmoittaneet laitteidensa tuen ko. protokollalle. Idea kaikilla Southbound API -rajapinnan protokollilla kuitenkin on yhteinen – määritellä tapa, kuinka SDN -kontrolleri vuorovaikuttaa verkkolaitteiden edelleenlähetystason kanssa, jotta tietoverkko voi paremmin adaptoitua verkon muutoksiin dynaamisesti. Verkkolaitteen vuotauluun (*flow-table)* voidaan kuitenkin tehdä sisäisiä merkintöjä myös staattisesti. OpenFlow -protokollasta esittely tarkemmin luvussa 2.5. (What are SDN Southbound APIs? n.d.)

2.4.4 Pohjoissuuntainen rajapinta

Kuvion 4 mukaan kontrollerilta pohjoiseen tulee sovellustaso Northbound API -rajapinnan kautta. Sovellustason tehtävä, kuten jo aikaisemmin on todettu, on luoda logiikka, jonka mukaan kontrolleri jakaa päätökset ja toiminnallisuudet verkkolaitteille edelleenlähetyksen suhteen. Sovellustason eri sovellukset ja palvelut mahdollistavat koko tietoverkon tehokkaan orkestroinnin ja automatisoinnin (What are SDN Northbound APIs? n.d.)

Tämä rajapinta on todennäköisesti kriittisin API -rajapinta SDN -ympäristössä, sillä ympäristön on tuettava näitä innovatiivisia sovelluksia, joihin koko SDN -ympäristön arvot ovat sidottu. Koska ne ovat niin kriittisiä ja ns. ”one size fits all” -ratkaisu ei ole realismia, pohjoissuuntaisen API -rajapinnan on pakko tukea laajasti erilaisia sovelluksia. Tämä yhtälö tekee tästä rajapinnasta SDN -ympäristön todennäköisesti epämääräisimmän komponentin – useita mahdollisia rajapintoja useisiin eri paikkoihin, joiden avulla hallitaan useita eri sovelluksia SDN -kontrollerin kautta (What are SDN Northbound APIs? n.d.)

Tänä päivänä tälle rajapinnalle ei ole vastaavaa standardia, kuten eteläsuuntaiselle rajapinnalle luotu OpenFlow. ONF -organisaatio on kuitenkin kirjoitushetkellä luonut ryhmän (*Northbound Working Group)*, joka on keskittynyt kehittämään pohjoissuuntaiselle API -rajapinnalle standardin, jonka tavoitteena on yksinkertaistaa verkkosovellusten käyttämistä niin, etteivät ne olisivat riippuvaisia tietystä kontrollerista tai palvelusta (What are SDN Northbound APIs? n.d.)

2.5 OpenFlow

2.5.1 Yleistä

OpenFlow on ensimmäinen virallinen standardoitu protokolla SDN -verkon laitteiden ja kontrollerin väliselle keskustelulle. Standardin loi Open Networking Foundation -organisaatio vuonna 2009, ja siitä asti sitä on kehitetty organisaation jäsenten puolesta. Protokolla on todella joustava, ja verkkolaitteiden keskitetty hallinta kontrollerille onnistuu huolimatta siitä, onko verkkolaite fyysinen vai virtuaalisesti toteutettu. Ennen OpenFlow:ta avoimen rajapinnan puute laitteiden edelleenlähetystasolle on johtanut siihen, että perinteinen tietoverkko on hyvin monoliittinen ja suljettu. Myöskään muut standardoidut protokollat eivät ole näin laajasti tuettu kuin OpenFlow. OpenFlow-verkkolaite (esim. kytkin) voi olla täysin OF-kytkin, jolloin se on tehty puhtaasti SDN-verkkoon tai sitten se voi olla kytkin, joka vain tukee OpenFlow-protokollaa, jolloin se voidaan kytkeä myös perinteiseen Ethernet-verkkoon. Tämä Ethernet-verkkoon luotu mutta OpenFlow:ta tukeva verkkolaite on hyödyllinen erityisesti silloin, kun ollaan siirtymävaiheessa perinteisestä tietoverkosta SDN-verkkoon ja koko infrastruktuuri ei vaadi välitöntä uudistusta. (ONF: Software-Defined Networking 2012, 9.)

2.5.2 Hyödyt

Varsinkin suuryritykset ja palveluntarjoajat hyötyvät OpenFlow -pohjaisesta SDN-ympäristöstä sen dynaamisuuden ja joustavuuden takia, jolloin hetkittäiset tarpeet nostaa vaatimuksia voivat tapahtua nopeasti. Lisäksi se selvästi vähentää prosesseissa välivaiheita ja monimutkaisuutta. (ONF: Software-Defined Networking 2012, 9.)

Suuri hyöty saadaan myös siitä, että OpenFlow -pohjaisessa verkossa voi olla usean eri valmistajan fyysisiä ja virtuaalisia laitteita ja koko orkestrointia voidaan hallita laitteiden valmistajasta ja mallista riippumatta keskitetysti, kunhan laite itsessään tukee kyseistä protokollaa.

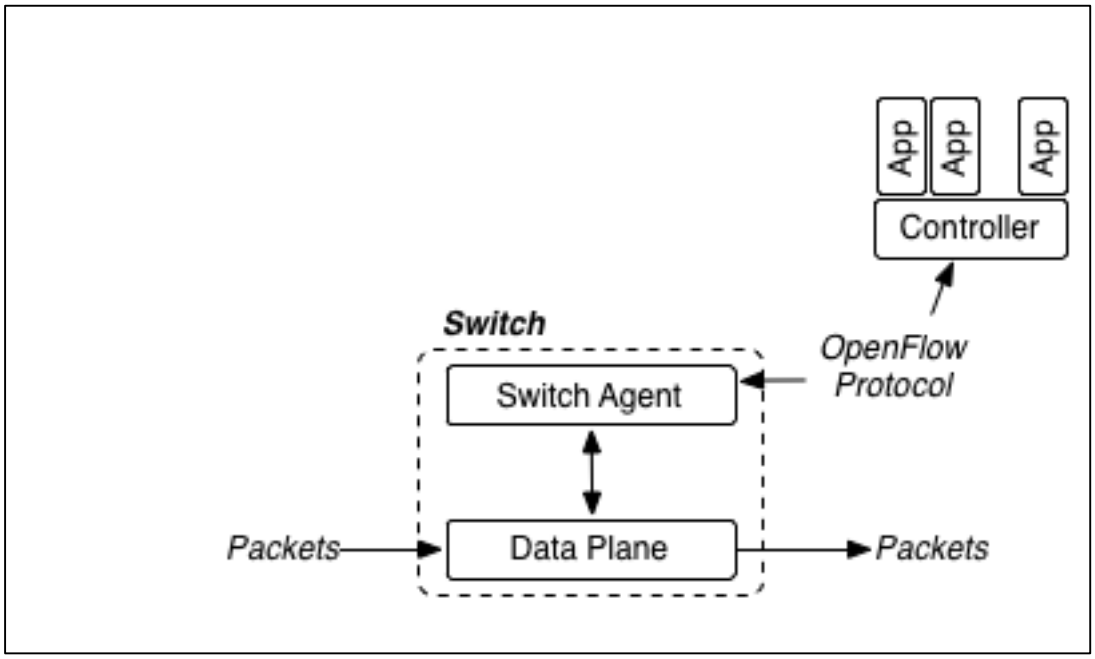
Joustavuudesta kertoo myös se, että SDN -verkossa on mahdollista ohjelmoida sovellustasolla koko tietoverkkoa koskevia asioita, mikäli sille nähdään tarvetta reaaliajassa. Tällainen muutos perinteisessä tietoverkossa on vaikeaa, mikäli edes mahdollista. (ONF: Software-Defined Networking 2012, 10.)

Tietoturvan suhteen työmäärä kevenee reilusti, koska sääntöjä ja politiikkoja voidaan jakaa yhdestä paikasta koko verkolle, jolloin vältytään yksittäisten laitteiden konfiguraatiosta. Tämä säästää reilusti välivaiheita verrattuna perinteiseen tietoverkkoon, jossa yhden laitteen lisäys tai poisto saattoi vaatia ACL -listojen muokkaamista usealla eri laitteella manuaalisesti. OpenFlow -pohjaisen SDN -verkon sisällä voidaan varmistaa tietoturva, palvelunlaatu ja pääsynhallinta koko verkossa – mukaan lukien langattomat verkot, sivukonttorit, kampukset ja datakeskukset (ONF: Software-Defined Networking 2012, 11.)

Erityisesti palveluntarjoajat voivat hyödyntää OpenFlow-pohjaisessa keskitetyssä hallinnassa hierarkista rakennetta, jolloin pääsynhallintasääntöjä voidaan sitoa esim. tiettyyn yhteyteen, käyttäjään, laitteeseen tai sovellukseen. Tämä antaa palveluntarjoajille yksinkertaisemman tavan ylläpitää verkkoa, jossa samaa infrastruktuuria voi käyttää useampi eri vuokralainen tai asiakkuus, jolloin verkon eristäminen elastisesti ja tietoturvallisesti on ensisijaisen tärkeää. (ONF: Software-Defined Networking 2012, 11.)

2.5.3 OpenFlow-kytkimen rakenne

OpenFlow -pohjaisessa SDN -verkossa, kuten muillakin protokollilla, kytkin prosessoi paketit sen mukaan, millaiset ohjeet se saa kontrollerilta. Jos tarkan arkkitehtuurin erilliset ohjelmointirajapinnat (esitelty luvussa 2.3) jätetään hetkeksi sivuun, voidaan OF -kytkimen anatomiassa elementti jakaa karkeasti kahteen komponenttiin: *Switch Agent* ja *Data Plane*. Tämä jako käy ilmi kuviossa 5. (Flowgrammable, 2016).



Kuvio 5. OpenFlow -kytkimen anatomia (Flowgrammable 2016)

Switch Agent keskustelee OpenFlow-protokollalla SDN -kontrollerin kanssa. Switch-Agent kääntää kontrollerilta tulevat komennot ja säännöt ikään kuin ”omalle kielelle” OpenFlow :n omalla sisäisellä protokollalla, joka on muilla protokollilla luonnollisesti omanlaisensa. Keskustelu tapahtuu myös toiseen suuntaan, jolloin Data Plane voi lähettää omia viestejään ensin agentille käyttämällä samaa sisäistä protokollaa, jonka jälkeen se Switch Agent -komponentin kautta kulkee OpenFlow -protokollalla SDN –kontrollerille (Flowgrammable 2016.)

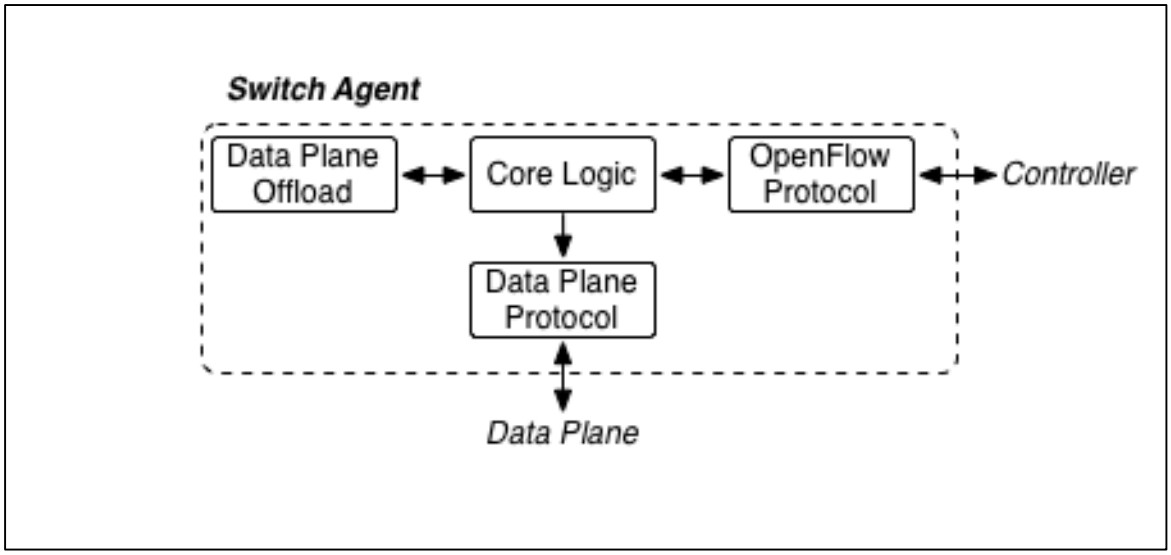
Data Plane koostuu seuraavista tiedoista: portit, vuotaulut, vuot, luokat ja toiminnot. Näihin tietoihin vaikuttamalla SDN -kontrolleri konfiguroi kytkimet toimimaan vastaantulevien pakettien suhteen halutulla tavalla (Flowgrammable 2016.) Vuotauluista enemmän luvussa 2.5.4.

Kuviossa 6 aikaisemmin mainittu Switch Agent -komponentti jaetaan tarkennuksen nimissä edelleen neljään komponenttiin, jotka ovat:

- **OpenFlow Protocol** – Kytkimen puoleinen rajapinta OpenFlow -yhteydessä kontrollerin kanssa.
- **Core Logic** – Hallitsee kytkintä, saattaa voimaan komennot kontrollerilta

- **Data Plane Offload** – Usein kontrolleritasolla purkautuneita toiminnallisuuksia OpenFlow :ssa, joita ei kuitenkaan implementoida Data Plane -tasolle.

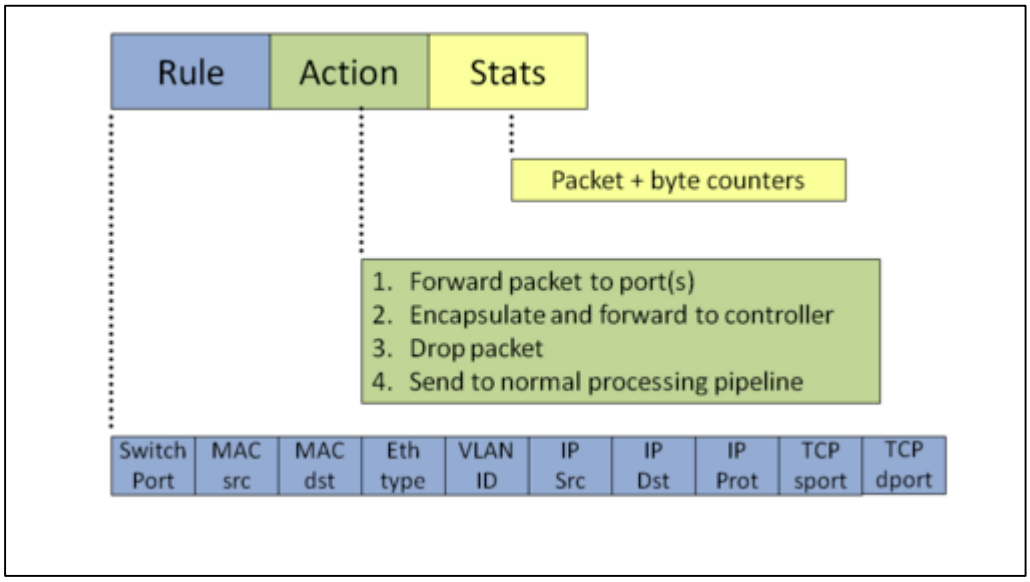
- **Data Plane Protocol** – Sisäinen protokolla ja niin sanottu ”kieli”, jolle Switch Agent -kääntää kontrollerin komennot, jotta ne voidaan ottaa edelleenlähetystasolla käyttöön (Flowgrammable, 2016).



Kuvio 6. OpenFlow -kytkimen agentin anatomia (Flowgrammable 2016.)

2.5.4 Vuotaulut ja pakettien prosessointi

OpenFlow -kytkimen hallinnasta eristetty edelleenvälitystaso siis toimii saapuvien pakettien suhteen, kuten SDN -kontrolleri on niille keskitetysti ohjeistanut. Tietyn reitin tai polun kulkema paketti muodostaa vuon (*Flow*). Vuomerkinnät tallennetaan kytkimen vuotauluun (*Flow-table*), joka pitää sisällään jokaisen polkutiedon ja säännöt, miten ko. polun paketit välitetään eteenpäin. Kuviossa 7 näkyy havainnollistavasti alkiot, joita manipuloimalla voidaan vaikuttaa vuotauluun.

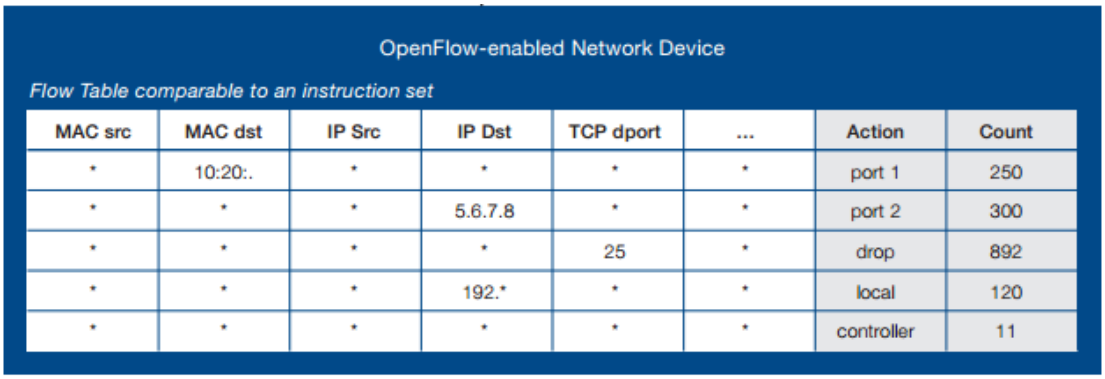


Kuvio 7.Vuotaulun manipuloitavat alkiot (McKeown Group, 2010)

Kuviossa sääntökentässä (*Rule*) on alkioita, joiden perusteella paketti liitetään tiettyyn vuohon. Kuten kuviosta 7 nähdään, voidaan kytkimelle saapunut paketti liittää tiettyyn vuohon tietyn portin, MAC -osoitteen tai vaikkapa VLAN ID :n perusteella(McKeown Group 2010.)

Seuraavaksi nähdään toimintakentässä (*Action)* neljä eri tapaa toimia saapuneen paketin suhteen. Päätös toiminnasta tulee sen mukaan, mihin vuohon paketti liittyy. Toimintoja on esim. välittää paketti tietystä portista ulos, välittää paketti kontrollerille tai pudottaa paketti. Toiminta on hyvin samantyylistä, kuin staattisissa ACL -listoissa, jossa toiminta perustuu siihen, mistä paketti tulee ja mihin se on menossa (McKeown Group 2010.)

Viimeisessä eli tilastokentässä (*Stats)* nimensä mukaisesti pidetään kirjaa pakettien ja prosessoitujen tavujen määrästä jokaista vuota kohti. Lisäksi kenttä huomioi ajat, jolloin tiettyä vuota koskeva viimeisin välityspäätös on tehty, jonka myötä voidaan tarvittaessa poistaa epäaktiiviset vuot. Hyvä esimerkki vuotaulusta näkyy kuviossa 8 (McKeown Group 2010.)



OpenFlow-enabled Network Device

Flow Table comparable to an instruction set

MAC src	MAC dst	IP Src	IP Dst	TCP dport	...	Action	Count
*	10:20:.	*	*	*	*	port 1	250
*	*	*	5.6.7.8	*	*	port 2	300
*	*	*	*	25	*	drop	892
*	*	*	192.*	*	*	local	120
*	*	*	*	*	*	controller	11

Kuvio 8. OpenFlow -kytkimen vuotaulu (ONF: Software-Defined Networking, 2012)

Kuvion 8 esimerkkivuotaulua voidaan tulkita käyttämällä hyväksi ylempänä selitettyjä sääntö-, toiminta- ja tilastokenttiä. Kuviosta nähdään, että jos kytkimelle saapuvan paketin kohde on 10:20: -alkuinen MAC -osoite, kuuluu se sellaiseen vuohon, jossa kytkin välittää sen ulos portista 1. Vastaavasti jos vastaanotetun paketin kohde on TCP -portti 25, kuuluu se vuohon, jossa paketti kuuluu pudottaa pois. Tarvittaessa vuotauluun voidaan myös asettaa tiettyjä prioriteetteja, jos saapuvassa paketissa on ristiriita. Tätä prioriteettia kuvaa kuvion 8 Count -kenttä. Ristiriitatilanteessa kytkin toimii sen mukaan, kummassa vuomerkinnän toiminnassa on korkeampi prioriteetti.

3 Langattoman lähiverkon perusteet

3.1 Yleistä

802.11 on IEEE-organisaation vuonna 1997 laatima standardi langattomille lähiverkoille. Langattomasta lähiverkosta käytetään lyhennettä *WLAN* joka tulee sanoista *Wireless Local Area Network* ja se muistuttaa teknisesti hyvin paljon perinteistä Ethernet-verkkoa, jonka standardi on 802.3. Erona perinteiseen Ethernet-verkkoon on, että WLAN käyttää siirtotiehen langatonta radioaaltoa, jolloin signaalit etenevät 2,4GHz tai 5,0GHz taajuusalueilla (Vähä-Touru 2007, 11-12.)

Langattoman verkon tiedonsiirto on alttiimpi virheille ja häiriöille, kuin langallinen Ethernet-tekniikka. Lähetyksen laatu riippuu käytetystä standardista, ilmatien esteistä ja ennen kaikkea välimatkasta laitteiden välillä. Lisäksi erityisesti vanhemmilla protokollilla WLAN-tekniikan heikompi tietoturva on ollut riskitekijä. Uhkia WLAN-verkoissa on aktiivisia ja passiivisia. Aktiivisia uhkia ovat esimerkiksi palvelunestohyökkäykset, jolla tulvitetaan verkkoa suurella määrällä palvelupyyntöjä tai välistävetohyökkäyksiä, joissa hyökkääjä asettaa laitteensa asiakaslaitteiden ja tukiasemien väliin päämääränä kaapata liikennettä. Passiivisiin uhkiin voidaan lukea erilaiset salakuuntelut ja analysoinnit. Näiden avulla pyritään selvittämään muun muassa verkon salausavaimet. On siis oleellista, että WLAN-verkon liikenne salataan jollain menetelmällä (*WEP, WPA, WPA2*) ja lisäksi toteutetaan autentikaatiomenetelmä, jolla voidaan todentaa verkkoon liittyvän käyttäjän tai laitteen identiteetti (Vähä-Touru 2007, 20-21.)

3.2 Keskeisimmät standardit

IEEE :n 802.11-standardista on luotu useita lisästandardeja, jotka laajentavat standardia lisäominaisuuksilla. Alkuperäisen 802.11-standardin maksimitiedonsiirtonopeus oli 2 Mbit/s ja sen käyttötaajuus oli 2,4 GHz. Tiedonsiirtoon käytettiin useita eri menetelmiä, kuten taajuushyppelyä (*FHSS*), suorasekvenssitekniikkaa (*DSSS*) ja infrapunatekniikkaa (Holm 2014, 9.)

IEEE 802.11b-standardi julkaistiin vuonna 1999 ja sen myötä tiedonsiirron maksiminopeus kasvoi arvoon 11 Mbit/s. Tekniikkana ainoastaan suorasekvenssitekniikka ja taajuusalueena 2,4 GHz. (Holm 2014, 10.)

IEEE 802.11a-standardi julkaistiin edellisen kanssa samaan aikaan ja se mahdollisti 5,0 GHz taajuusalueen käyttämisen ja 54 Mbit/s tiedonsiirtonopeuden. Tekniikkana tuotiin käyttöön myös OFDM- eli monikantoaaltomodulointi. OFDM-modulaatiossa lähetettävä data jaetaan alikanaviin, jotka ovat eri taajuuksilla ja niitä käytetään rinnakkain. 5,0 GHz taajuusalueen etuna on pienempi määrä häiriöttömiä kanavia, kuin 2,4 GHz taajuusalueella, mutta korkeamman taajuuden myötä kantama ja peittoalue ovat pienempiä (Holm 2014, 11.)

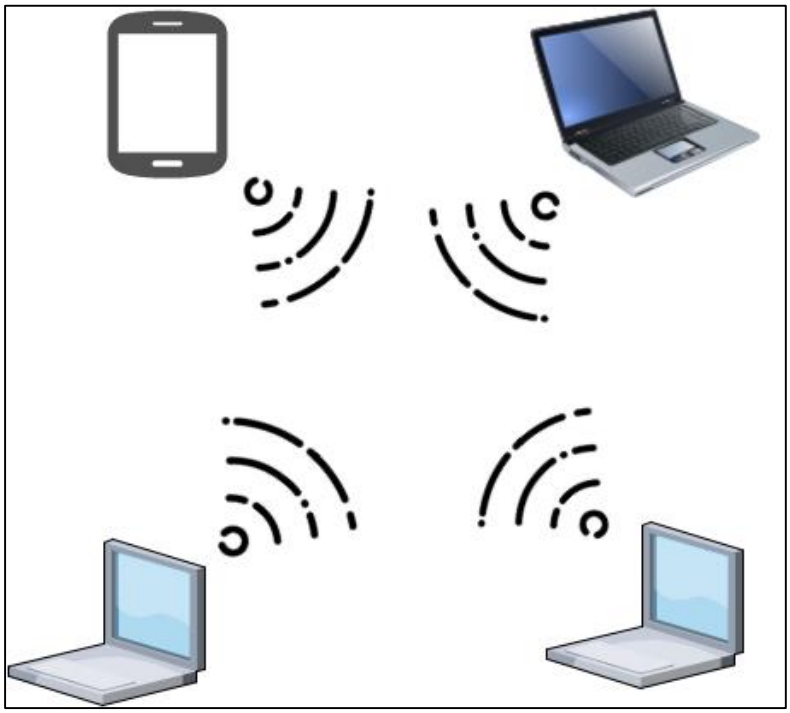
IEEE 802.11g julkaistiin vuonna 2003 ja on taaksepäin sopiva b-standardin kanssa, sillä ne käyttävät samaa 2,4 GHz taajuutta. Tiedonsiirron maksiminopeus edelleen 54 Mbit/s ja tekniikkana OFDM-modulaatio. Standardia pidetäänkin yhdistelmänä hyödyntäen b-standardin isompaa peittoaluetta ja a-standardin tiedonsiirtonopeutta (Holm 2014, 11.)

IEEE 802.11n on 2009 julkaistu standardi. Se toi käyttöön MIMO-tekniikan, joka tulee sanoista *Multiple-Input Multiple-Output*. Kuten sanoista voidaan päätellä, käytetään tekniikassa useampaa antennia sekä signaalin lähettämiseen että vastaanottamiseen. Tämän ja suuremman kanavakohtaisen kaistanleveyden (*40MHz)* myötä teoreettinen tiedonsiirron maksiminopeus nousi arvoon 600 Mbit/s. Tekniikassa voidaan käyttää sekä 2,4 että 5,0 GHz-taajuusalueita (Holm 2014, 12.)

IEEE 802.11ac on standardeista uusin ja se hyväksyttiin vuonna 2014. Se on laajennus erityisesti n-standardista, vaikkakin taajuusalueena käytetään ainoastaan 5,0 GHz:n kaistaa. Laajennus tukee *Multi-User* MIMO:a, joka poikkeaa tavallisesta MIMO:sta niin, että käytettäviä antenneja on enemmän. Lisäksi kanavakohtaisia kaistanleveyksiä tuetaan useita erilaisia: 20, 40, 80 ja 160 MHz. Suurimmalla kanavakohtaisella kaistanleveydellä (160MHz) voidaan yhdellä antennilla saavuttaa n. 866 Mbit/s tiedonsiirtonopeus, joka tarkoittaa MU-MIMO tekniikan kahdeksalla antennilla jopa 6,7 Gbit/s teoreettista tiedonsiirron maksiminopeutta (LitePoint 2013).

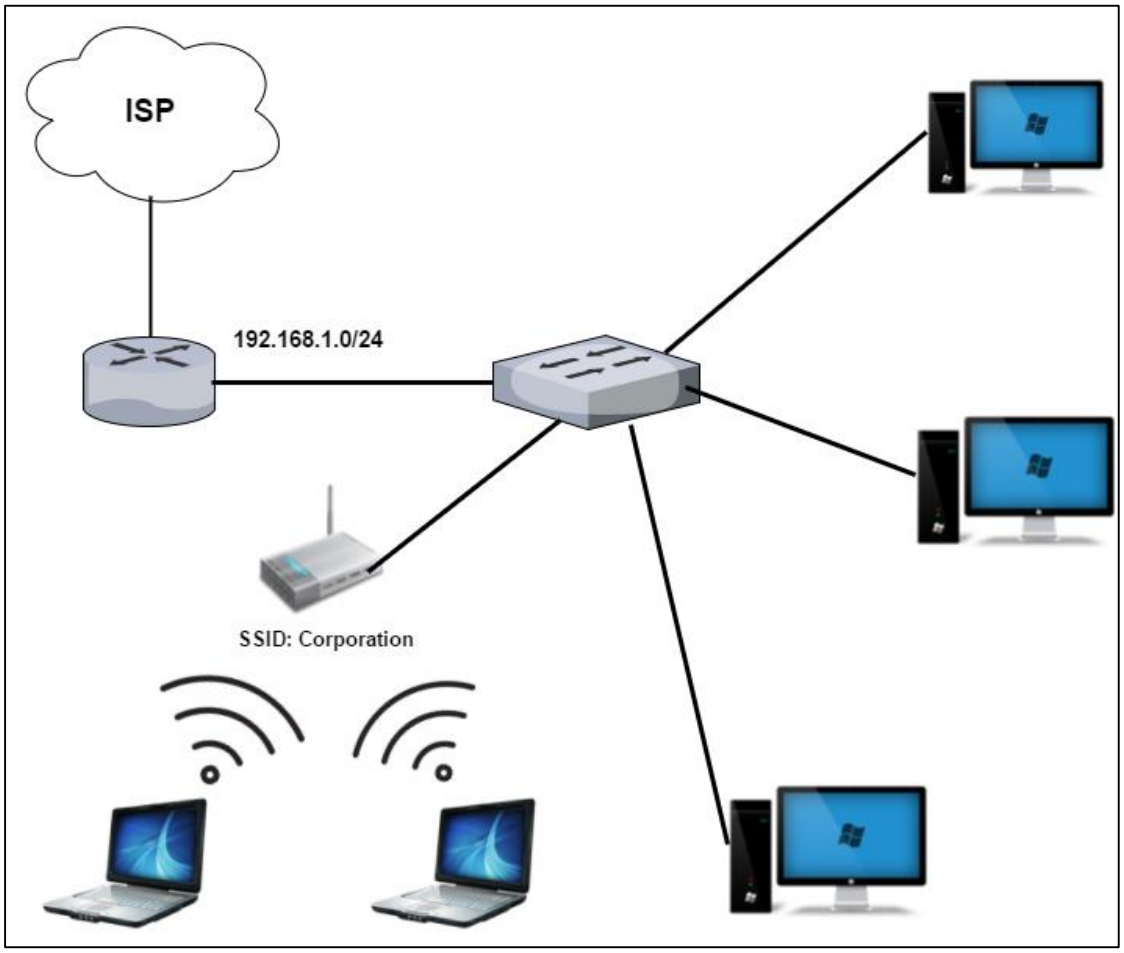
3.3 Topologiat

WLAN-tekniikan topologiat voidaan karkeasti jakaa kahteen: *Ad-Hoc*-topologia ja *infrastruktuuri*-topologia. Ad-Hoc-topologia koostuu ainoastaan asiakaslaitteista (kannettavat tietokoneet, älypuhelimet, tabletit yms), jotka kommunikoivat keskenään suoraan ilman tukiasemaa. Topologiaa käytetään yleensä vain paikallisesti pienessä työryhmässä, vaikkakin pääsy esim. Internetiin on mahdollinen jakaa joltain topologian laitteelta. Kuviossa 9 on esitelty tämä topologia (Vähä-Touru 2007, 15.)



Kuvio 9. Ad-Hoc topologia

Infrastruktuuri-topologiassa oleellinen komponentti on langaton tukiasema (*Access Point, AP*), jonka tehtävä on liittää langattomat asiakaslaitteet olemassa olevaan langalliseen Ethernet-verkkoon. Tukiasema on tällöin siltaavassa tilassa, joten tukiasema ei tee langattomille laitteille omaa aliverkkoa, vaan ne ovat langallisten asiakaslaitteden tapaan samassa aliverkossa yhteydessä reitittimeen ja tämän kautta Internetiin. Topologia esitetty kuviossa 10 (Värä-Touru 2007, 16.)



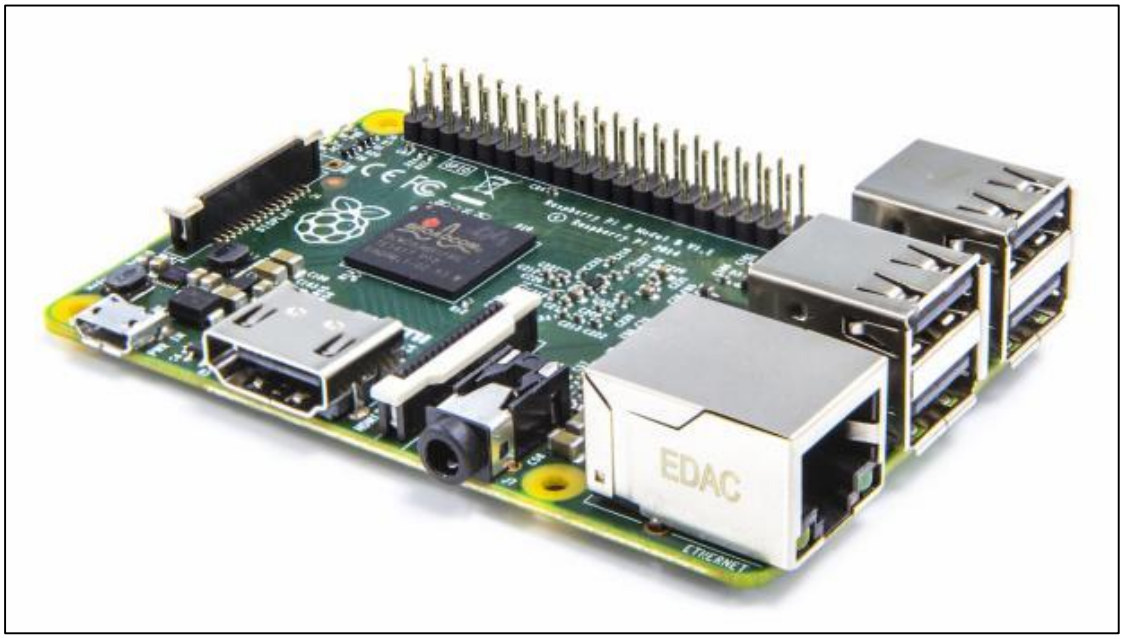
Kuvio 10. Infrastruktuuri-topologia

4 Käytetyt laitteet ja ohjelmat

4.1 Raspberry Pi

Toteutusvaiheessa langattoman verkon tukiasemat tuli rakentaa Raspberry Pi-piirilevytietokoneista. Raspberry Pi on *The Raspberry Pi Foundation* -organisaation kehittämä kämmentä pienempi piirilevy, jossa on kaikki tietokoneen vaatimat komponentit ja liitännät. Heidän tavoitteenaan oli edistää koulutusta ja harrastuneisuutta tietokoneisiin ja toiveena oli, että tämä edullinen ja pieni työkalu mahdollistaa sen. Raspberry Pi :stä on jo julkaistu useampi eri malli, mutta tässä työssä valittiin uusin 3.0, sillä se on toistaiseksi ainoa, jossa on rajapinta langattomaan tiedonsiirtoon (802.11 b/g/n). Liitteessä 1 on esitetty kokonaisuudessaan kaikki Raspberry pi 3.0 :n tekniset tiedot (Verry 2012.)

Raspberry Pi :hin tuli asentaa avoimen lähdekoodin käyttöjärjestelmä Micro SD -massamuistikortille, jotta tukiaseman konfigurointi voitiin aloittaa. Käyttöjärjestelmän valinta ja sen tiedot on esitetty omassa kappaleessaan. Kuviossa 11 esitellään Raspberry Pi 3.0 :n ulkoinen olemus.



Kuvio 11. Raspberry Pi 3.0

4.2 Raspbian OS

Raspbian OS on avoimen lähdekoodin käyttöjärjestelmä Raspberry Pi -tietokoneille, joka perustuu Linuxin Debian -jakelupakettiin. Käyttöjärjestelmä on hieman riisuttu tavallisesta työaseman versiosta, mutta sisältää silti mm. kaikki perusohjelmat, työkalut ja yli 35 000 erikseen ladattavaa pakettia. Ensimmäinen versio järjestelmästä julkaistiin vuonna 2012, mutta siitä asti sitä on kehitetty aktiivisesti käyttäjien toimesta. Käyttöjärjestelmä ja sen asennettavat paketit ovat ilmaisia, mutta sen kehityksestä vastaavat ihmiset saavat tukea toimintaansa satunnaisilta lahjoittajilta. Itse Raspberry Pi -tietokoneen kehittäjät ja Raspbianin kehittäjät eivät tee keskenään minkäänlaista yhteistyötä. (Welcome to Rasbian n.d.)

4.3 Open vSwitch

Open vSwitch on avoimen lähdekoodin ohjelmallisesti toteutettu monikerroksinen virtuaalikytkin. Se on suunniteltu erityisesti virtuaalisoituun palvelinympäristöön. Kytkin välittää liikennettä eri virtuaalikoneiden kanssa samalla fyysisellä isäntäkoneella ja siitä eteenpäin fyysiseen tietoverkkoon aivan, kuten fyysinenkin kytkin. OVS tukee laajasti erilaisia toiminnallisuuksia ja protokollia, kuten esimerkiksi OpenFlow :ta, jonka vuoksi OVS on tämän työn kannalta oleellinen vaihtoehto. Kaikki OVS :n tukemat toiminnallisuudet on nähtävillä liitteessä 2.

OVS on suunniteltu yhteensopivaksi modernien kytkimien piirisarjojen kanssa. Tämä tarkoittaa, että sen saa osaksi olemassa olevaa fyysistä kytkinverkkoa ja se sallii saman joustavan hallittavuuden, kuin fyysinenkin kytkinverkko. OVS -kytkintä voi ajaa kaikilla Linux -pohjaisilla virtualisointialustoilla, kuten KVM, VirtualBox, Xen, Xen Cloud Platform ja XenServer. Valtaosa lähdekoodista on kirjoitettu C -ohjelmointikielellä, joten OVS :ää on mahdollista soveltaa muihinkin ympäristöihin. (Open vSwitch n.d.)

4.4 Hostapd

Hostapd on Linux-käyttöjärjestelmille toteutettu ”daemon”, jolla voidaan toteuttaa tukiasema, jotta langattomat lähiverkon asiakaslaitteet saadaan liitettyä perinteiseen Ethernet-verkkoon. Hostapd tukee tarvittaessa tekstipohjaisia frontend-käyttöliittymiä, mutta tässä työssä käytettiin oletuksena löytyvää CLI:tä (*Command Line Interface)*. Hostapd on daemon, joka tarkoittaa, että se on suunniteltu käynnistymään taustalle prosessina, joten se ei vaadi jatkuvaa ja suoraa hallintaa. Daemonit myös saadaan usein käynnistymään samalla, kun itse laitteet uudelleenkäynnistyvät, joten ne eivät senkään puolesta vaadi erillistä manuaalista toimenpidettä (Malinen 2013.)

Vaikkakin tässä opinnäytetyössä langattomasta verkosta tehdään avoin, niin Hostapd tukee useita erilaisia tapoja (802.1X/WPA/WAP/RADIUS) autentikoida verkkoon liittyvät käyttäjät, joka edistää tietoturvaa. Kaikki autentikointiin liittyvät tuetut ominaisuudet on esitelty liitteessä 3. Hostapd tukee useita erilaisia langattomia verkkokortteja ja ajureita, joista tässä työssä oleellisimmat olivat *nl80211/cfg80211*-ajurit. Ohjelman mukana käyttöjärjestelmälle asentuu konfiguraatiotiedosto (*hostapd.conf)*,

jonka parametreja muokkaamalla saadaan konfiguroitua ominaisuuksilta haluttu langaton verkko. (Malinen 2013.)

4.5 Bridge-utils

Bridge-utils on erittäin kevyt ohjelma Linux-käyttöjärjestelmille, jonka tarkoituksena on tehdä laitteista siltaavia. Ohjelma on käyttöjärjestelmässä nimellä *brctl* ja se voidaan kuvitella virtuaalikytkimeksi, jossa niin ikään luodaan Bridge-laite, kuten OVS:llä. Tähän virtuaaliseen Bridge-laitteeseen lisätään rajapinnat, jotka halutaan sillata, jonka jälkeen tämän takana olevat asiakaslaitteet ovat samassa aliverkossa isompaan tietoverkkoon. Brctl on helpoin kuvitella tavalliseksi ei-hallittavaksi L2-tason Ethernet-kytkimeksi, joka on ohjelmallisesti toteutettu ja siihen liitettäviä rajapintoja voidaan itse muokata. Bridge-utilsissa oli lisäksi valmis integraatio Hostapd:n kanssa, joka vaikutti merkittävästi siihen, miksi tällaiseen ratkaisuun päädyttiin (Bridging Network Connections 2016.)

4.6 PfSense

PfSense on valmis ohjelmistopalomuuri. Se perustuu Unixin kaltaisiin käyttöjärjestelmiin (FreeBSD) ja on avoimen lähdekoodin tuotteena täysin ilmainen. Se voidaan asentaa suoraan fyysiselle raudalle tai virtuaalikoneelle. PfSense sisältää kaikki samat tärkeät ominaisuudet, kuin fyysiset ja kaupalliset palomuurit, joten se kelpaa täysin esimerkiksi tuotantoverkkoihin. Kaikki PfSensen palomuuraukseen ja reititykseen liittyvät ominaisuudet on listattu liitteessä 4 (Introducing pfSense 2016.)

4.7 ONOS

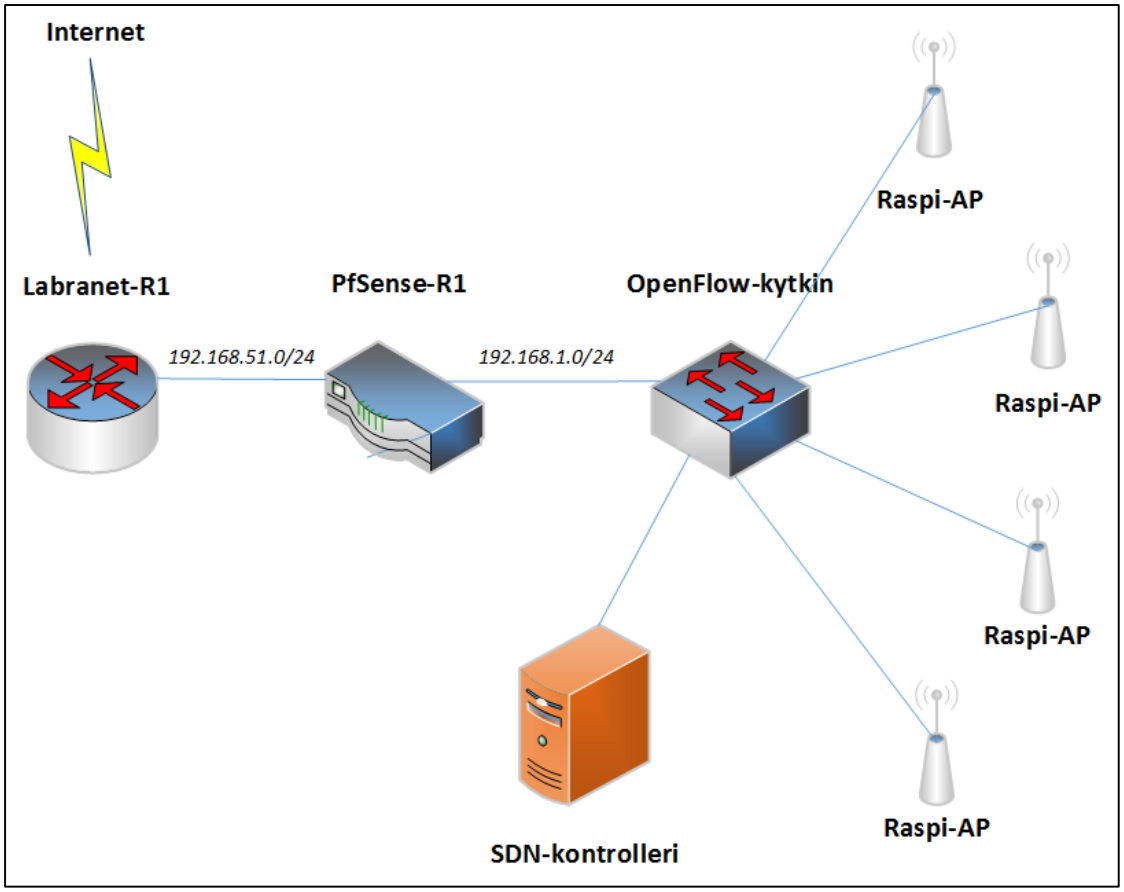
ONOS eli *Open Network Operating System* on avoimen lähdekoodin Linux-pohjainen käyttöjärjestelmä SDN-kontrollerille. ONOS on useiden merkittävien palveluntarjoajien ja laitevalmistajien yhteistyössä luotu järjestelmä, joten se on myös suunnattu jopa aitoon operaattoritason käyttöön. ONOS tarjoaa vikasietoisuutta, skaalautuvuutta ja suorituskykyä. Avoimen lähdekoodin SDN-kontrollerivaihtoehtoja oli

useita, mutta tämä valittiin työhön sen takia, että JAMK:n CyberTrust-hankkeen projektityöntekijöillä siitä oli käytännön kokemusta, joten siitä löysi merkittävästi hyvää sisäistä dokumentaatiota (ON.LAB 2014.)

5 Suunnitelma

5.1 Looginen topologia

Koska työn tavoitteena oli saada SDN -verkkoon WLAN -ratkaisu, pyritään verkon niin sanotusta ”kiinteästä” osasta tekemään yksinkertainen. Työn kokonainen looginen topologia SDN-testiverkosta ja yhteydestä ulkoverkkoon on esitetty kuviossa 12. Realistinen hyöty SDN -verkosta saadaan, kun kytkinverkko on laaja ja verkko sisältää muitakin kontrollerin kautta hallittavia verkkoelementtejä, mutta tässä opinnäytetyössä ei todettu järkeväksi monimutkaistaa topologiaa, jotta saavutetaan työn alkuperäiset tavoitteet.



Kuvio 12. SDN -verkon looginen topologia

Kuvion laitteista SDN-kontrolleri ja PfSense toteutettiin tässä opinnäytetyössä virtuaalisesti. Niiden konfigurointi ja dokumentaatio tuli kuitenkin suorittaa niin, että kun ko. verkko tulevaisuudessa implementoidaan fyysisillä laitteilla tuotantoon, voidaan niiden asennus suorittaa seuraten tämän dokumentin työvaiheita ja ohjeita. Eli työn suunnitelmaa laatiessa oli otettu huomioon se, että verkko tullaan tekemään fyysisillä laitteilla, mutta käytännön syistä osa komponenteista on virtualisoitu, jotta työtä voitiin tehdä mahdollisimman paljon samassa luokkatilassa.

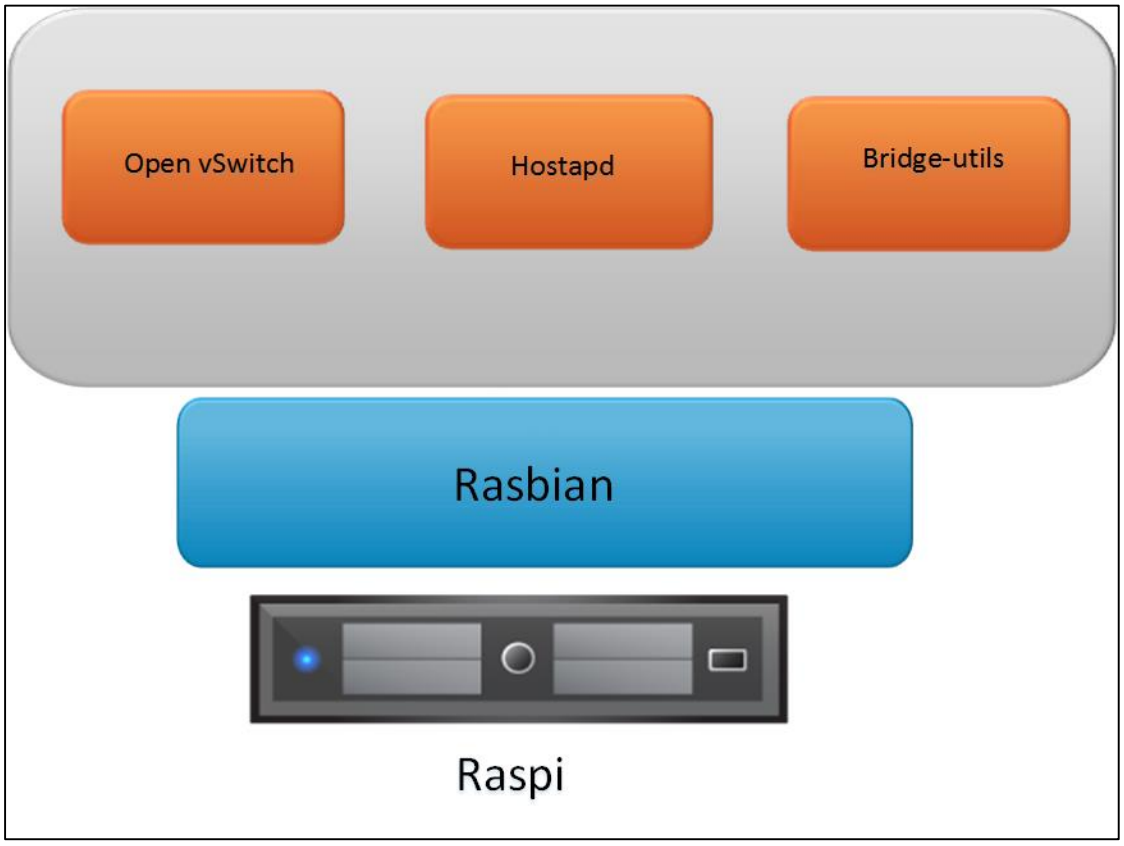
Yhteys ulkoverkkoon muodostuu LabraNetin reitittimen kautta. Koska SDN-testiverkko halutaan omaan aliverkkoon, tulee LabraNetin reitittimen jälkeen asentaa reitittävä laite, joka tässä työssä on PfSense-ohjelmistopalomuuri. PfSensen WAN-puolen rajapinta saa osoitteensa DHCP:llä LabraNetin osoiteavaruudesta ja LAN-puolen rajapinnalle laite tekee uuden aliverkon. PfSense on kaapeloitu OpenFlow-kytkimeen. OpenFlow-kytkimeen kaapeloidaan kiinni WLAN-tukiasemat, jotka tuli rakentaa käyttämällä Raspberry Pi 3.0-piirilevytietokoneita. Kytkimeen tulee myös virtuaalisesti toteutettu SDN-kontrolleri, jonka avulla SDN-testiverkon verkkoelementtejä tulee pystyä hallitsemaan.

SDN-kontrollerilla on hallintayhteys fyysiseen tai virtuaaliseen OpenFlow-kytkimeen ja langattomiin tukiasemiin OpenFlow-protokollan avulla. Saman rajapinnan kautta kulkee myös asiakaslaitteiden hyötydataliikenne. Kontrolleritason ja edelleenvälitystason liikenteet on eristetty toisistaan, vaikka ne jakavat saman fyysisen rajapinnan OpenFlow-kytkimen ja tukiasemien välillä.

Langattomien asiakaslaitteiden liittyessä SDN-verkkoon langattoman rajapinnan kautta, tulee heille aueta selaimen avulla ilmoitus verkon PfSense-palomuurilta, jossa kerrotaan, että heidän yhteyttään tullaan monitoroimaan ja käyttämään tutkimustyössä osana CyberTrust-projektia. Tämä laskeutumissivu, jolle käyttäjät pakotetaan, jotta yhteys voidaan muodostaa, on termiltään *Captive Portal*. Kun nämä ehdot on luettu, voidaan ne hyväksyä, jonka jälkeen yhteys verkkoon sallitaan. Kirjautumiseen on mahdollista lisätä autentikaatio usealla eri tavalla.

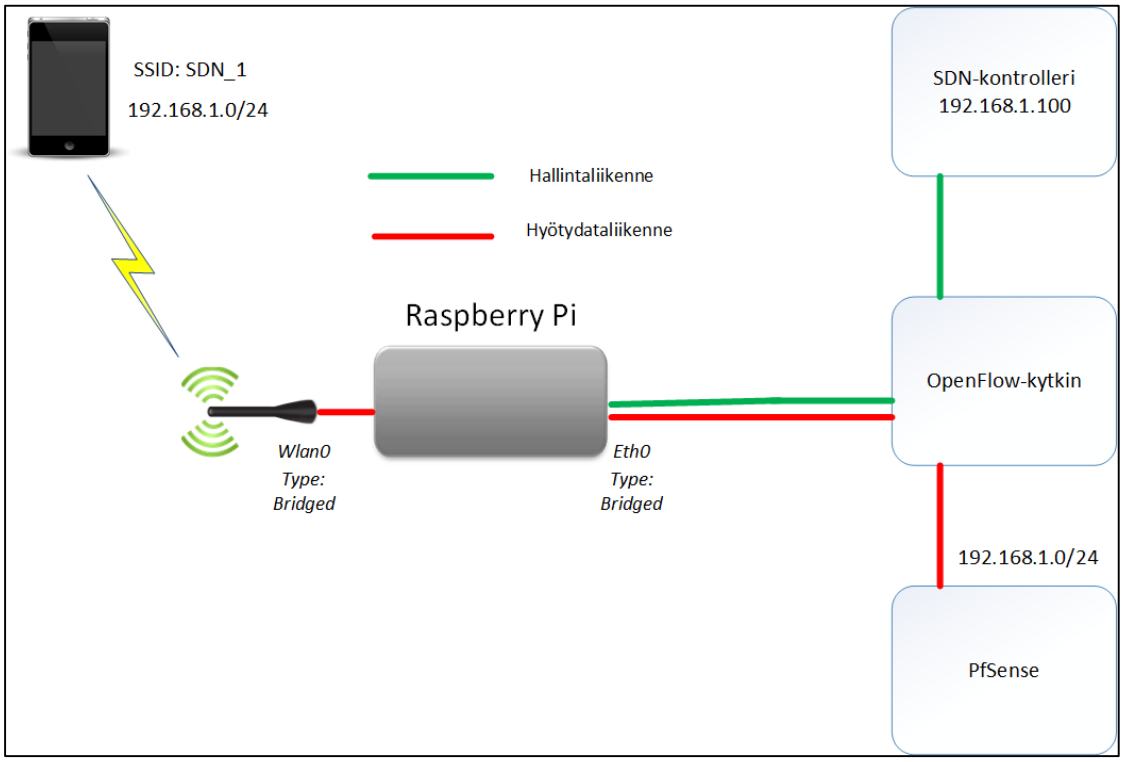
5.2 Tukiasemat

Tukiasemien asennus aloitettiin asentamalla Raspbian käyttöjärjestelmä Raspberry Pi-tietokoneelle. Kun käyttöjärjestelmään päästiin kirjautumaan sisään, asennettiin Raspbianin päälle Open vSwitch, Bridge-utils ja Hostapd. Nämä komponentit on havainnollistettu kuvioon 13. OVS-virtuaalikytkin konfiguroidaan niin, että se toimii normaalin kytkimen sijaan WLAN-tukiaseman ja OpenFlow-kytkimen yhdistelmänä.



Kuvio 13. Tukiaseman arkkitehtuuri

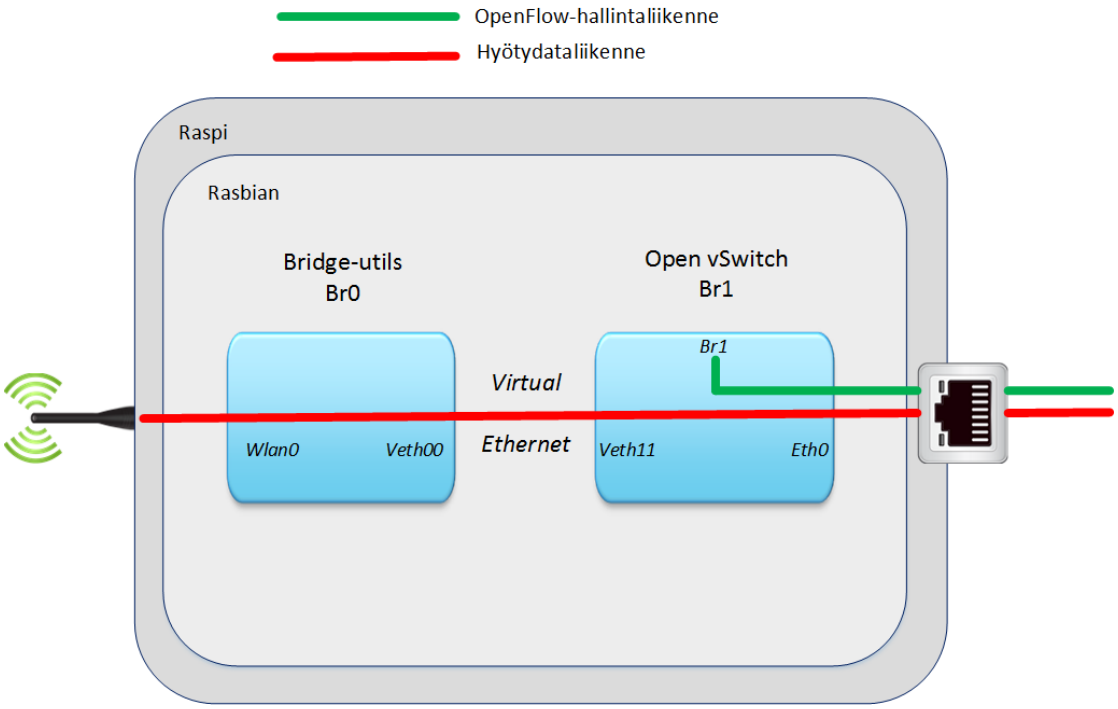
Jotta Raspista saatiin tehtyä tehtyä WLAN-tukiasema, täytyi sen fyysiset rajapinnat *eth0* ja *wlan0* konfiguroida niin, että ne olivat siltaavassa tilassa, jotta se saatiin PfSensen tekemään verkkoon eikä tukiasema tee edelleen uutta aliverkkoa. Tällöin samaan virtuaaliseen Bridge-kytkimeen tulee molemmat hyötydataliikenteen rajapinnat eli *wlan0* ja *eth0*. Rajapinnat sisältävä rakennekaavio on esitetty kuviossa 14.



Kuvio 14. Tukiaseman virtuaalikytkimen Br0-rajapinnat

Raspbianin päälle tulee kaksi eri ohjelmalla tehtyä virtuaalikytkintä. Koska riittää, että hallintayhteys saadaan laitteen yhdelle rajapinnalle, voidaan rajapinta *wlan0* liittää Bridge-utilsilla luotuun virtuaalikytkimeen Br0 siltaavassa tilassa. Rajapinta *eth0* sen sijaan liitetään OVS:n Br1-virtuaalikytkimeen, joka mahdollistaa samalla OpenFlow-protokollan tuonnin Raspille. Nämä kaksi virtuaalikytkintä yhdistetään toisiinsa virtuaalisella Ethernet-yhteydellä (*Virtual Ethernet)*. Nämä virtuaaliset Ethernet-rajapinnat *veth00* ja *veth11* liitetään virtuaalikytkimiin ja niiden välinen linkki nostetaan ylös laitteen käynnistyessä.

Seuraavaksi tuli ottaa huomioon OpenFlow-protokollan tuonti Raspille. Tukiaseman hallinta suunniteltiin *In-Band* -mallilla, jolloin hallinnalle tarkoitettu rajapinta tulee tehdä OVS:n Br1-kytkimen virtuaaliselle oletusrajapinnalle, joka on tyypiltään *Internal*. Tämä rajapinta konfiguroidaan erillisellä TCP –yhteydellä SDN-kontrollerille. Virtuaalikytkinten ja hallintayhteyden looginen topologia esitetty kuviossa 15.

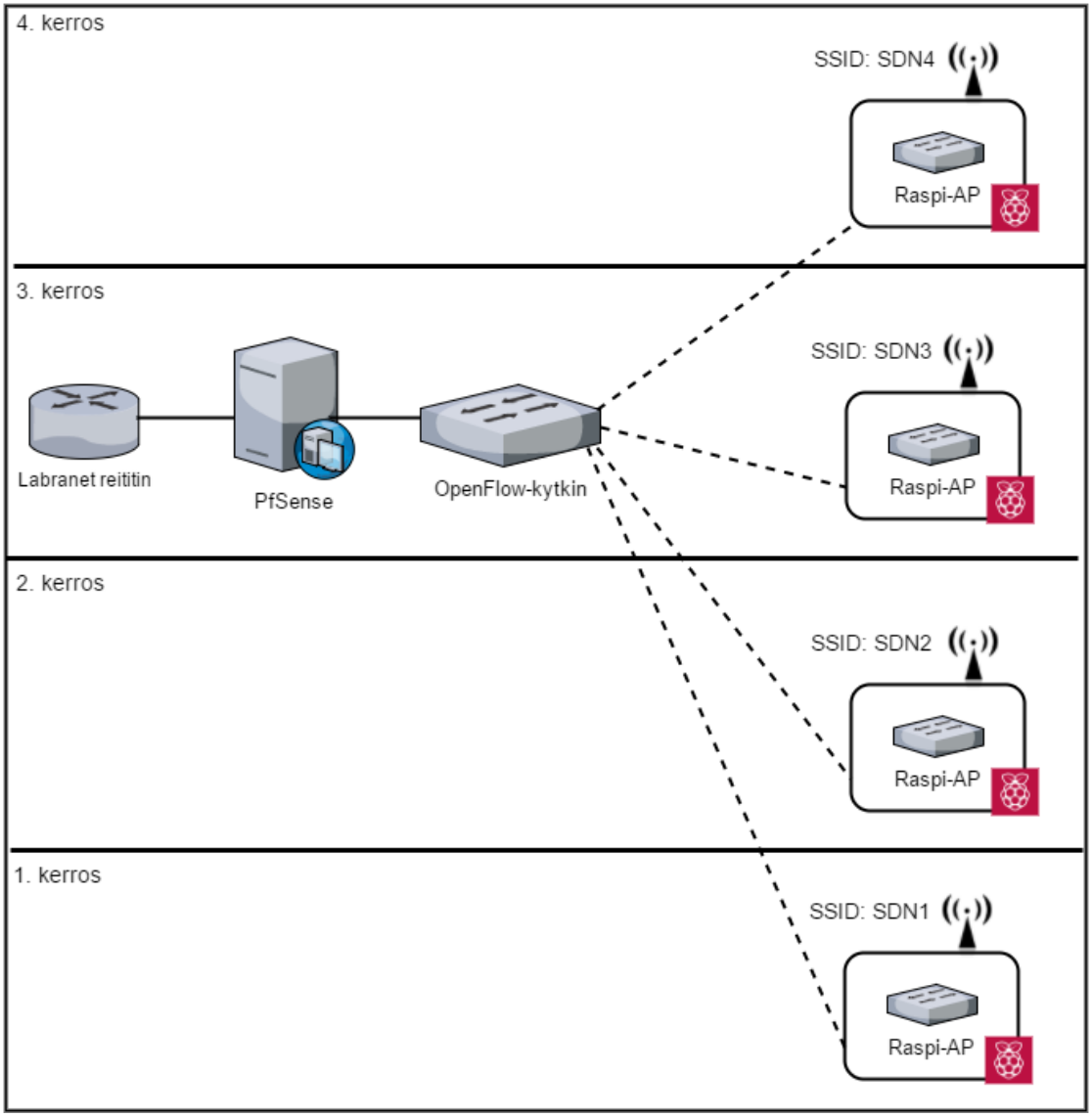


Kuvio 15. Virtuaalikytkinten br1 ja br0 looginen topologia

OpenFlow-protokollan hallintaliikenne siis kulkee laitteen OVS:n Br1 –rajapinnalle samaa fyysistä kaapelia pitkin, kuin hyötydataliikenne, mutta yhteys SDN-kontrollerille on eristetty virtuaalisesti hyötydataliikenteestä.

5.3 Tukiasemien sijoitus

Kun virtuaalisesti simuloitu ympäristö tulevaisuudessa tuodaan fyysisillä laitteilla tuotantoverkkoon, on tukiasemat tarkoitus sijoittaa Jyväskylän Ammattikorkeakoulun Dynamon toimipisteessä kuvion 16 mukaisesti.

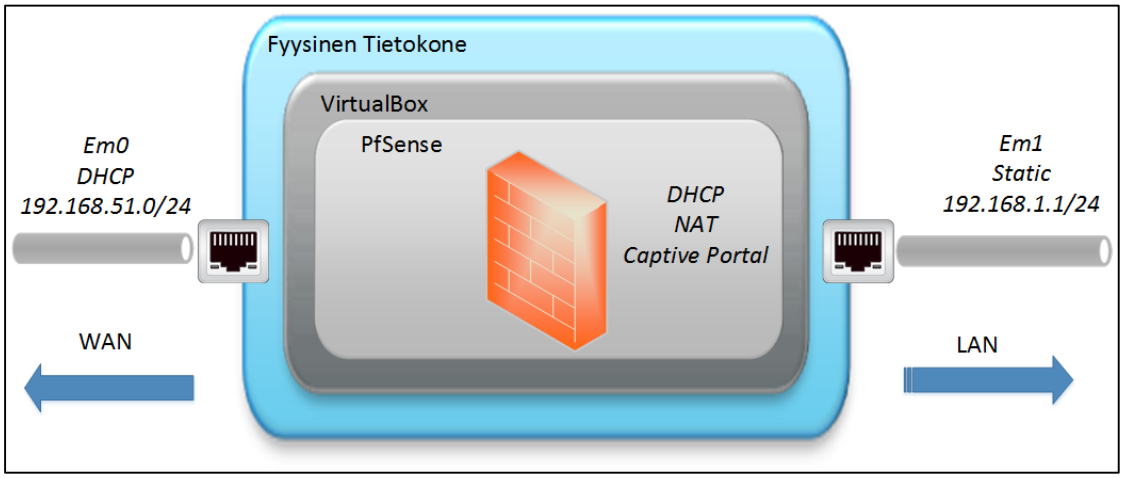


Kuvio 16. Tukiasemien fyysinen topologia

Kuten kuviosta näkee, niin jokainen langaton tukiasema on omalla SSID :llä luotu. Tämä siitä syystä, että työn tekohetkellä ei ollut vielä tutkittu lainkaan, onko Raspberry Pi-tietokoneilla ja Open vSwitch-virtuaalikytkimillä mahdollista edes toteuttaa ns. ”roaming” -ominaisuutta, jossa langattomat päätelaitteet osaavat asiakkaan liikkuessa automatisoidusti yhdistää uuteen ja lähempänä olevaan tukiasemaan samalla SSID :llä, kun nykyisen tukiaseman signaali heikkenee merkittävästi. Käytännössä kaikki tukiasemat ovat siis samalla mallilla tehtyjä. Tukiasemia toteutetaan useita sen takia, että saadaan enemmän hyötydataa langattomista asiakaslaitteista, joita voidaan tutkia myöhemmin.

5.4 Palomuuri

SDN-testiverkon ja LabraNetin väliin tuleva palomuuri toteutetaan PfSensellä. Palomuuri on virtualisoitu Virtualbox-virtualisointialustalla. Fyysinen tietokone, jolla virtualisointi toteutetaan, on kaapeloitu WAN-puolelta LabraNetin suuntaan ja LAN-puolelta SDN-verkon suuntaan (kts. kuvio 12). Tämä edellyttää fyysiseltä tietokoneelta kaksi verkkoadapteria. Tämä on havainnollistettu kuviossa 17.



Kuvio 17. PfSensen looginen rakenne

6 Toteutus

6.1 Tukiasemien asennus ja konfigurointi

6.1.1 Tukiaseman komponenttien ja palveluiden asennus

Tukiaseman asennus aloitettiin lataamalla verkosta Rasbian-käyttöjärjestelmän ISO-image, joka kirjoitettiin 8Gt–kokoiselle MicroSD-kortille. Tämä kortti asennettiin Raspiin, jolloin se voitiin käynnistää ko. käyttöjärjestelmään. Toistaiseksi Raspi on kiinni verkkokaapelilla LabraNetiin, jotta se on verkkoyhteydessä tarvittavia asennuksia varten. Samalla Raspi sai LabraNetistä IP-osoitteen, jolloin saatiin Raspiin SSH-yhteys Putty-ohjelmalla, joten tulevien komentojen kopioiminen dokumentaatioon oli helpompaa. Rasbian käyttöjärjestelmä oli Kerner-versiolla 4.4.21-v7+. Ensimmäiseksi asennettiin Raspille Open vSwitch seuraavilla komennoilla.

root@raspberrypi:/home/pi# apt-get update

root@raspberrypi:/home/pi# apt-get install -y autoconf libtool openssl pkg-config make gcc libssl-dev

Sparse jouduttiin lataamaan tässä vaiheessa käsin, koska sitä ei löydy nykyisistä repositorioista.

root@raspberrypi:/home/pi# git clone git://git.kernel.org/pub/scm/devel/sparse/sparse.git

root@raspberrypi:/home/pi# cd sparse

root@raspberrypi:/home/pi/sparse# make

root@raspberrypi:/home/pi/sparse# make install

Nyt Sparse oli asennettu, jonka jälkeen voidaan kloonata OVS Raspille, jonka jälkeen määritetään kansiot konfiguraatioille ja *boot.sh*-skriptille. Huomioitava kansion vaihto tässä vaiheessa.

root@raspberrypi:/home/pi/sparse# cd ..

root@raspberrypi:/home/pi# git clone https://github.com/openvswitch/ovs.git

root@raspberrypi:/home/pi# cd ovs/

root@raspberrypi:/home/pi/ovs# sudo apt-get install dh-autoreconf

root@raspberrypi:/home/pi/ovs# ./boot.sh

root@raspberrypi:/home/pi/ovs#./configure --prefix=/usr --localstatedir=/var --sysconfdir=/etc

Tässä vaiheessa järjestelmä herjasi, että Python-ohjelmointikielestä puuttuu moduuli *six*. Asennetaan käsin, jonka jälkeen *make*-komennolla käännetään lähdekooditiedostot ajettavaksi tiedostoksi.

root@raspberrypi:/home/pi/ovs# apt-get install python-pip

root@raspberrypi:/home/pi/ovs# pip install six

root@raspberrypi:/home/pi/ovs# make –j3

root@raspberrypi:/home/pi/ovs# make install

root@raspberrypi:/home/pi/ovs# cp debian/openvswitchswitch.init /etc/init.d/openvswitch-switch

OVS asennettu. Seuraavaksi voidaan käynnistää OVS ja samalla tarkastaa sen versio seuraavilla komennoilla. Näistä on esitetty tuloste kuviossa 18.

root@raspberrypi:/home/pi/ovs# /etc/init.d/openvswitch-switch start

root@raspberrypi:/home/pi/ovs# ovs-vsctl show

```
root@raspberrypi:/home/pi/ovs# /etc/init.d/openvswitch-switch start
Inserting openvswitch module.
/etc/openvswitch/conf.db does not exist ... (warning).
Creating empty database /etc/openvswitch/conf.db.
Starting ovsdb-server.
/usr/share/openvswitch/scripts/ovs-ctl: 1: /usr/share/openvswitch/scripts/ovs-ctl: uuidgen:
not found
missing uuidgen, could not generate system ID ... failed!
Configuring Open vSwitch system IDs.
Starting ovs-vswitchd.
Enabling remote OVSDB managers.
root@raspberrypi:/home/pi/ovs# ovs-vsctl show
b4a31ffc-9c1a-4400-ae90-9315f852bd4b
    ovs_version: "2.6.90"
```

Kuvio 18. OVS:n käynnistys ja version todennus

Kuten kuviosta voidaan todeta, on OVS:n versio 2.6.90. Lisäksi luotiin *ovsdb-server*, johon OVS:n konfiguraatiot tallennetaan ja OVS on käynnissä. Tämän jälkeen voitiin alkaa luomaan virtuaalikytkimiä suunnitelman mukaisesti (kts. kuvio 15). Ensimmäiseksi oli kuitenkin oleellisinta luoda Virtuaalisen Ethernet-linkin rajapinnat *veth00* ja *veth11*, koska ne liitetään jo alussa virtuaalikytkimiin. Sama komento myös luo niiden välille linkin. Tämä tapahtuu seuraavalla komennolla:

root@raspberrypi:/home/pi# ip link add veth00 type veth peer name veth11

Seuraavaksi luotiin Bridge nimellä *br1* ja lisätään siihen rajapinnat *eth0* ja *veth11* seuraavilla komennoilla. Todennus on esitetty kuviossa 19.

root@raspberrypi:/home/pi/ovs# ovs-vsctl add-br br1

root@raspberrypi:/home/pi/ovs# ovs-vsctl add-port br1 eth0

root@raspberrypi:/home/pi/ovs# ovs-vsctl add-port br1 veth11

```
root@raspberrypi:/home/pi# ovs-vsctl show
b4a31ffc-9c1a-4400-ae90-9315f852bd4b
    Bridge "br1"

        Port "veth11"
            Interface "veth11"
        Port "eth0"
            Interface "eth0"
        Port "br1"
            Interface "br1"
                type: internal
    ovs_version: "2.6.90"
```

Kuvio 19. Br1-niminen virtuaalikytkin ja sen rajapinnat

OVS:ssä portit ovat oletuksena siltaavassa tilassa, joten sen suhteen ei tarvinnut tehdä muutoksia. Seuraavaksi tuli asentaa ohjelma Bridge-utils, jolla voitiin tehdä suunnitelman (kts. kuvio 15) toinen virtuaalikytkin. Edelliset suoritettiin seuraavilla komennoilla:

root@raspberrypi:/home/pi# apt-get install bridge-utils

root@raspberrypi:/home/pi# brctl addbr br0

root@raspberrypi:/home/pi# brctl addif br0 wlan0

root@raspberrypi:/home/pi# brctl addif br0 veth00

Tämän virtuaalikytkimen ID ja siihen liitetyt portit voidaan todeta komennolla *brctl show br0*. Tämä kuviossa 20.

```
root@raspberrypi:/home/pi# brctl show br0
bridge name     bridge id               STP enabled     interfaces
br0             8000.b827ebc9fa50       no              veth00
                                                        wlan0
```

Kuvio 20. Br0-niminen virtuaalikytkin ja sen rajapinnat

Tämän jälkeen täytyi tukiasema liittää verkon SDN-kontrolleriin. Tämä suoritettiin seuraavalla komennolla OVS:n *br1*-kytkimen konfiguraatioon:

root@raspberrypi:/home/p/ovsi# ovs-vsctl set-controller br1 tcp:192.168.1.100:6633

Seuraavaksi asennettiin Debian-pohjainen WAP (*Wireless Access Point*)-ohjelma, jonka konfiguraatiotiedostoa muokkaamalla vaikutetaan tukiaseman teknisiin asetuksiin. Ohjelman asennus suoritetaan seuraavalla komennolla:

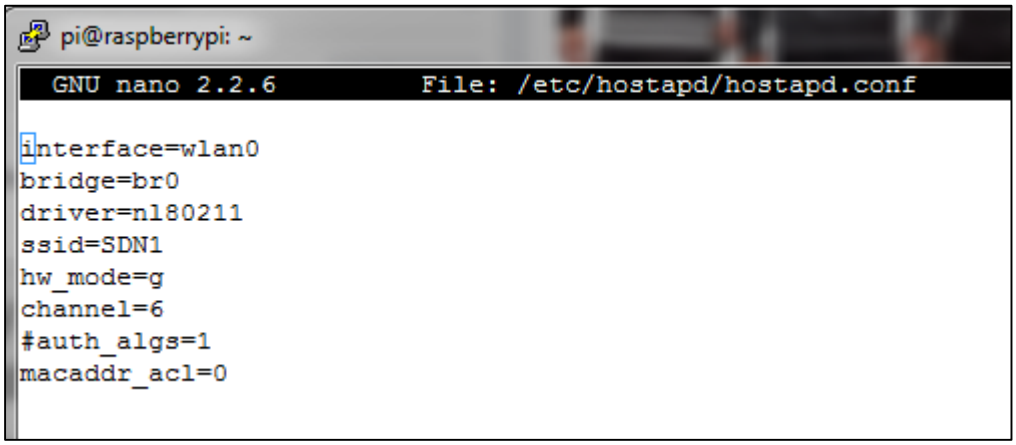
root@raspberrypi:~# sudo apt-get install hostapd

Asennus loi ohjelmalle tiedoston polkuun */etc/default/hostapd*. Nyt täytyi mennä tiedostoon tekstieditorilla ja muokata parametria *DAEMON_CONF* niin, että ko. daemon käynnistyessään lukee konfiguraatiot oikeasta tiedostosta. Tämän tiedoston sijainti käy ilmi kuviosta 21.

```
  GNU nano 2.2.6                File: /etc/default/hostapd

# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Kuvio 21. Hostapd.conf-konfiguraatiotiedoston sijainnin määritys

Luotiin ko. absoluuttiseen polkuun tekstitiedosto *hostapd.conf* halutuilla parametreilla. Tiedosto on esitetty kuviossa 22.



Kuvio 22. Hostapd.conf-tiedosto

Seuraavaksi käynnistettiin palvelu *hostapd* ja alustettiin sen *conf*-tiedosto seuraavilla komennoilla:

root@raspberrypi:/home/pi# /etc/init.d/hostapd start

[ok] Starting hostapd (via systemctl): hostapd.service.

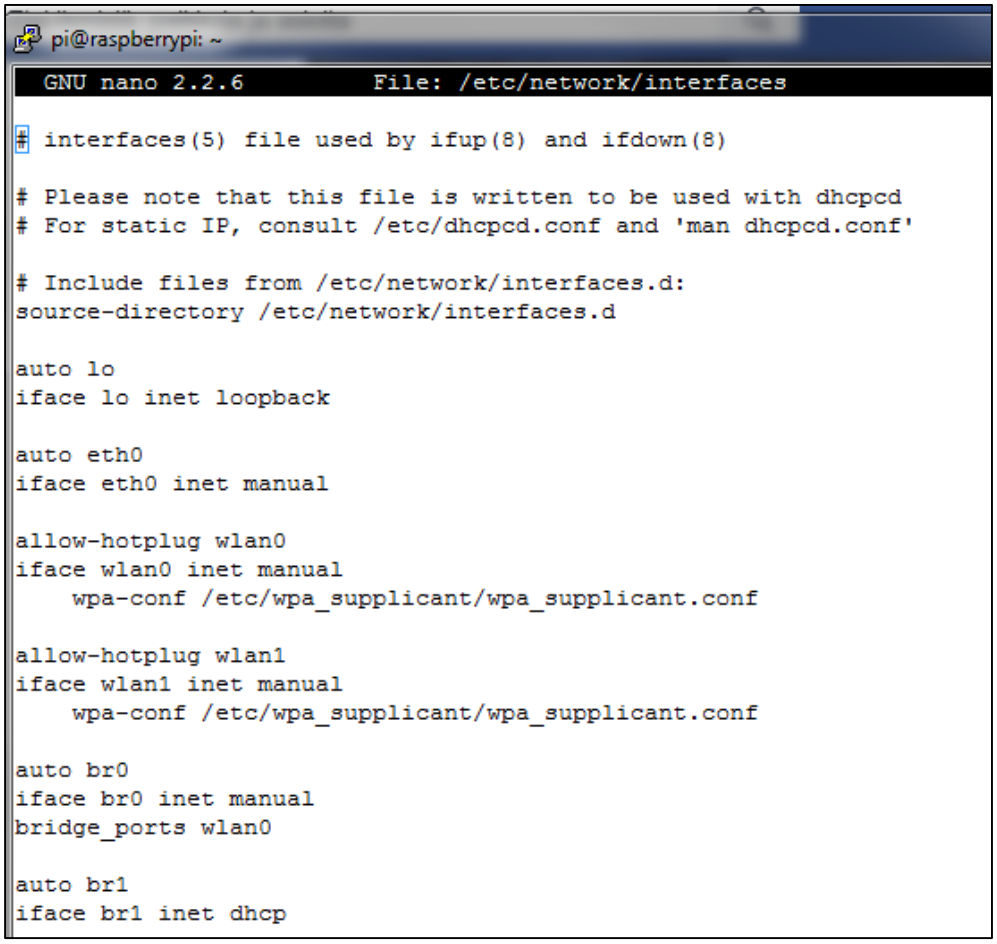
/usr/sbin/hostapd /etc/hostapd/hostapd.conf &

Tämän jälkeen komentokehote ilmoitti, että millä MAC-osoitteella wlan0 on asetettu ENABLED-tilaan halutulla SSID:llä. Tämän jälkeen voidaan todeta, että rajapinnan tyyppi on AP (*Access Point)* seuraavalla komennolla. Kahden edellisen toimenpiteen tulosteet on esitetty kuviossa 23.

```
root@raspberrypi:/home/pi# /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:c9:fa:50 and ssid "timonsdnverkko1"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
^Croot@raspberrypi:/home/pi#
root@raspberrypi:/home/pi# iw wlan0 info
Interface wlan0
        ifindex 3
        wdev 0x1
        addr b8:27:eb:c9:fa:50
        type AP
        wiphy 0
root@raspberrypi:/home/pi#
```

Kuvio 23. Wlan0-rajapinnan alustus ja tyypin todennus

Tämän jälkeen täytyi *interfaces*-tiedostoon määrittää esimerkiksi mitkä portit sillataan ja rajapinnat, joille halutaan määrittää IP-osoitteet dynaamisesti DHCP:llä tai staattisesti yms. Tämä on esitetty kuviossa 24.



Kuvio 24. /etc/network/interfaces-tiedosto

Näillä asennuksilla, kun tarvittavat palvelut ja virtuaalisen Ethernet-linkin nostaa ylös, liittää rajapinnan *veth00* kytkimeen *br0* ja alustaa hostapd.conf-tiedoston aiemmin mainitulla komennolla, niin haluttu langaton verkko mainostuu päätelaitteille ja siihen voidaan yhdistyä, mutta ongelmana on laitteen vikasietoisuus. Jos tukiasema menisi syystä tai toisesta virheelliseen tilaan, täytyisi Raspille kirjautua sisään fyysisesti paikan päällä ja asetuksia sekä palveluita nostamaan aktiiviseksi käsin.

6.1.2 Tukiaseman vikasietoisuuden parantaminen

Seuraavaksi oli aiheellista panostaa tukiaseman vikasietoisuuteen, jotta mahdollisessa vikatilanteessa laite voidaan vain käynnistää uudelleen, jonka jälkeen tarvittavat palvelut, rajapinnat ja komponentit nousevat aktiivisesti oikeassa järjestyksessä.

Tässä käytettiin hyväksi *crontab*-työkalua ja *rc.local*-tiedostoa. Crontab-työkalulla voidaan käynnistyksen yhteydessä suorittaa skriptejä, komentoja ja palveluita. Työkaluun määritetyt asetukset on esitetty kuviossa 25.

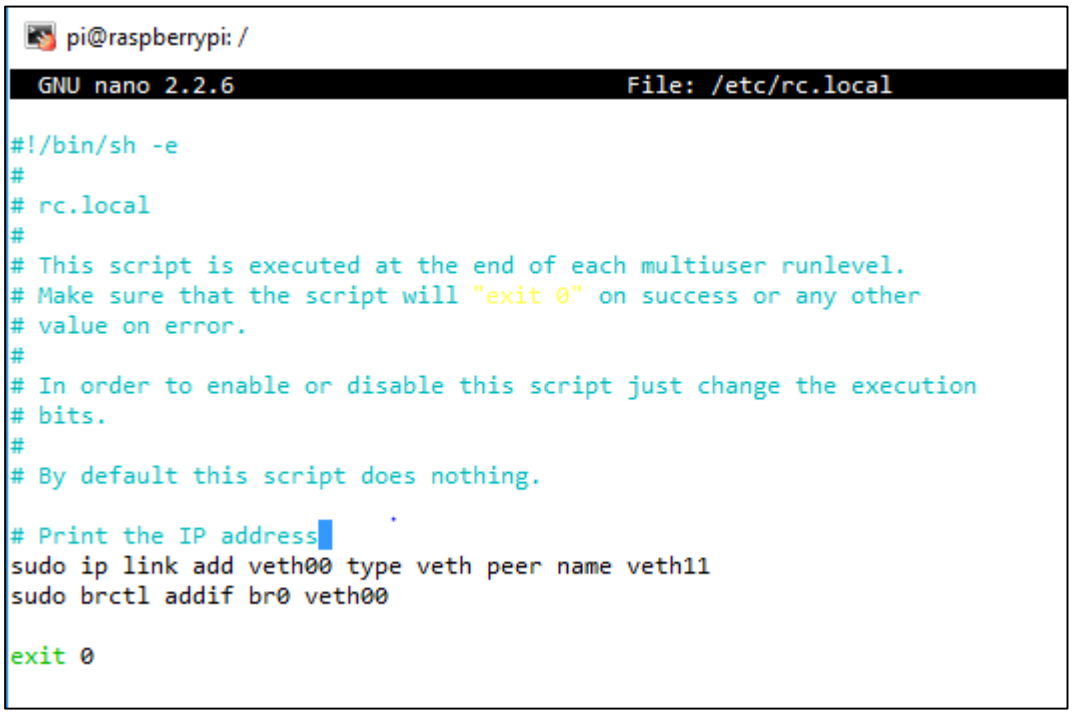
```
  GNU nano 2.2.6        File: /tmp/crontab.BOY7NT/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot sudo /etc/init.d/openvswitch-switch start
@reboot sudo /etc/init.d/hostapd start
@reboot sudo sleep 30 && sudo dhclient br1
@reboot sudo sleep 40 && sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf &
@reboot sudo sleep 45 && sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf &
```

Kuvio 25. Crontab-asetukset

Kuten kuviosta käy ilmi, määritys *@reboot* määrää, että seuraava komento suoritetaan, kun laite käynnistyy uudestaan. Asetuksiin määritettiin järjestyksessään käynnistymään Open vSwitch, Hostapd-palvelu sekä DHCP-pyyntö rajapinnalle *br1*. Viimeiseksi 40 ja 45 sekunnin viiveillä hostapd.conf-tiedoston alustus ja Raspin AP-moodi tilaan *Enabled*. Tämä viive sen takia, että tämä alustus tuli tehdä viimeisenä

myös manuaalisesti, joten viiveellä varmistetaan, että alustus tapahtuu viimeisenä myös automatisoidussa uudelleenkäynnistyksessä.

Rc.local-tiedosto taas on valmiina Linux-käyttöjärjestelmistä löytyvä skripti, joka ajetaan oletuksena käynnistysprosessin (*init)* loppuvaiheessa. Rc.local näin ollen suoritetaan vasta crontabin jälkeen. Rc.local-tiedostoa muokattiin kuvion 26 mukaisesti.



Kuvio 26. Rc.local-skripti

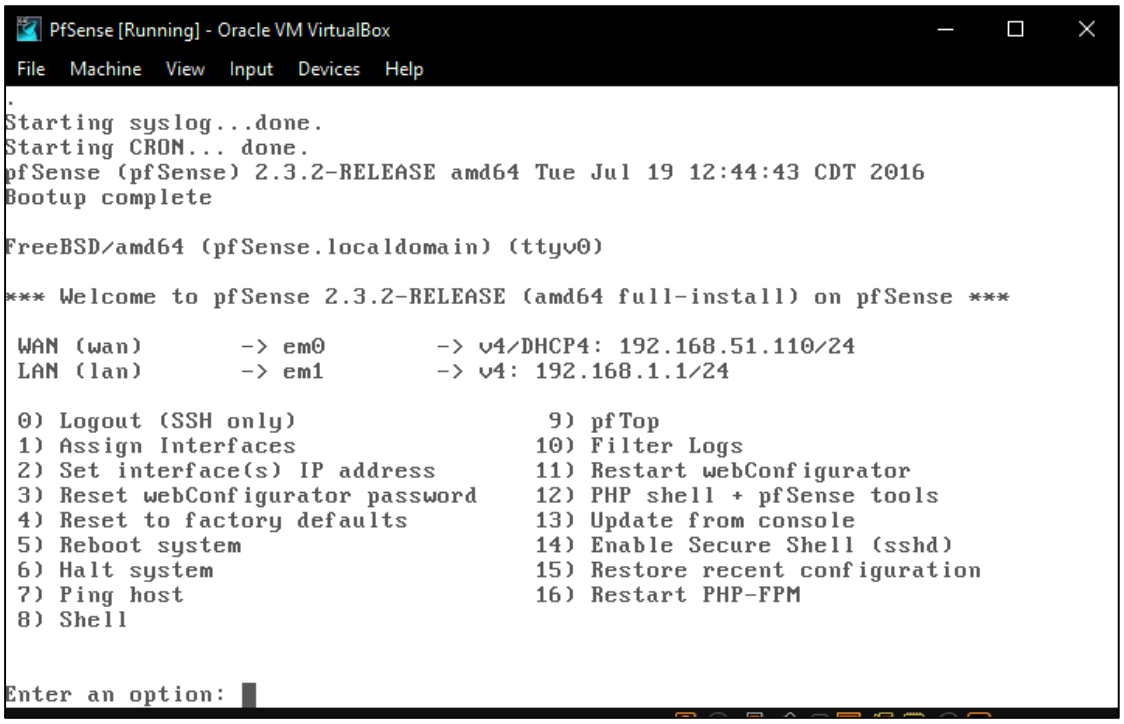
Kuten kuviosta näkyy, skripti luo virtuaalisen Ethernet-rajapinnan rajapintojen *veth00* ja *veth11* välille, jonka jälkeen liittää rajapinnan *veth00* virtuaalikytkimeen br0. Rajapintaa *veth11* ei tarvitse erikseen liittää virtuaalikytkimeen br1, koska tämän tiedot jäävät OVS:n konfiguraatioihin, vaikka laite uudelleenkäynnistetään.

Tämän jälkeen, kun mahdollisessa vikatilanteessa Raspista irrottaa virtajohdon ja liittää sen takaisin, nousevat palvelut ja rajapinnat oikeassa järjestyksessä ylös ja verkkoon liittyminen on mahdollista, kunhan yhteys SDN-kontrollerille on olemassa.

6.2 PfSense

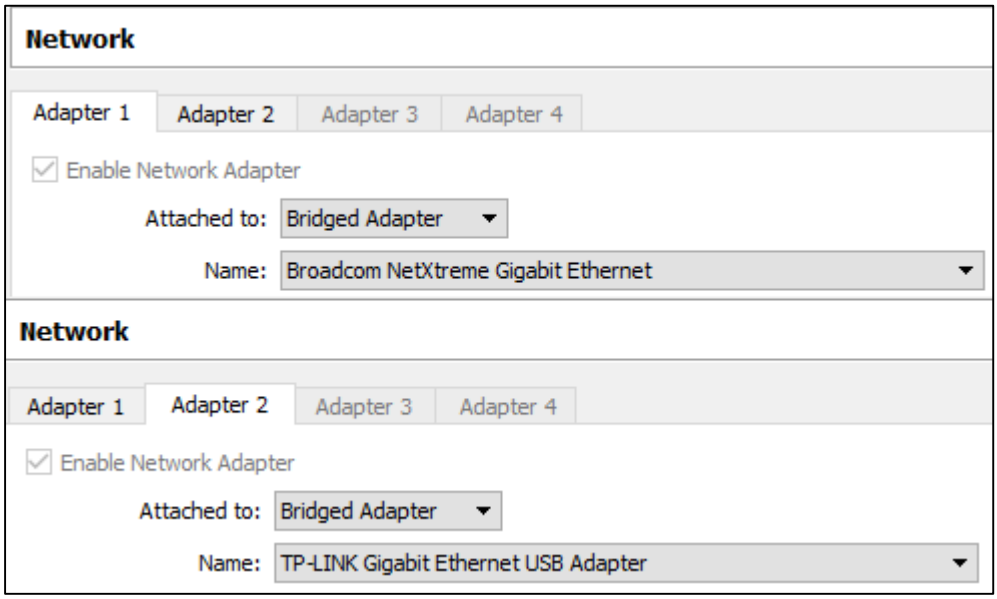
6.2.1 Rajapinnat

PfSensen asentaminen aloitettiin lataamalla ohjelman verkkosivuilta ISO-image 64-bittiselle AMD-käyttöjärjestelmälle. Virtualbox-virtualisointialustalla luotiin uusi virtuaalikone, jonka tyyppi oli FreeBSD. Tämän jälkeen kone voitiin käynnistää, jonka jälkeen asennus oli aika pitkälle automatisoitu. Asennuksen jälkeen virtuaalikoneen asetuksista tuli poistaa PfSensen Live-CD, jonka jälkeen uudelleenkäynnistys määritettiin palomuuriin. Kone käynnistyy yksinkertaiseen CLI-näkymään. Tästä näkymästä on esittely kuviossa 27.



Kuvio 27. PfSense-palomuurin CLI-käyttöliittymä

Kuten kuviosta näkee, hakee PfSense dynaamisesti WAN-rajapinnallensa DHCP:ltä osoitteen, joka on tässä tapauksessa LabraNetin osoiteavaruudesta. LAN-puolen rajapinta on oletuksena osoitteella *192.168.1.1*. Koska kone on virtualisoitu, tulee Virtualboxin asetuksissa olla koneelle kaksi verkkoadapteria, jotka molemmat ovat tyyppiä *Bridged*. Kuviossa 28 näkyy tarkemmat tiedot verkkoadaptereista.



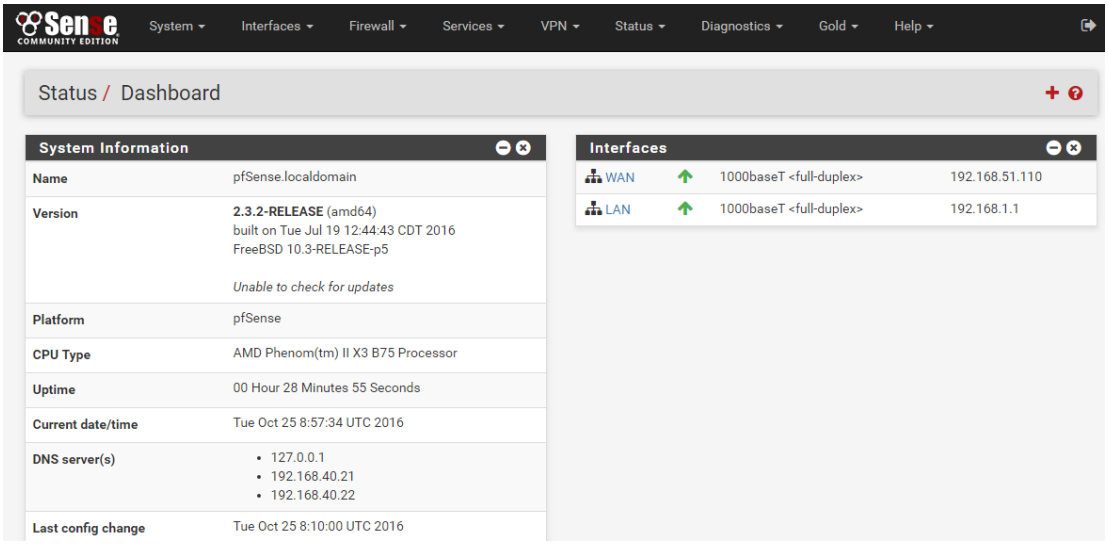
Kuvio 28. PfSense-virtuaalikoneen verkkoadapterit

Kuvion adapteri 1 on fyysinen tietokoneen oma verkkoadapteri, joka on sillattu LabraNetin suuntaan (WAN). Verkkoadapteri 2 sen sijaan on USB-porttiin liitettävä TP-LINK-merkkinen Ethernet-adapteri, joka on sillattu SDN-verkon suuntaan (LAN).

PfSensen konfigurointi ja hallinta onnistuu myös graafisella käyttöliittymällä selaimen kautta. Koska PfSensen LAN-puolella ei ollut vielä yhtään päätelaitetta, jouduttiin hallintayhteys ottamaan tilapäisesti WAN-rajapinnan kautta. Oletuksena tämä on lähes poikkeuksetta palomuureissa estetty, joten CLI:llä jouduttiin asettamaan seuraava komento, jolla sallittiin kyseinen toimenpide:

> *[2.3.2-release][root@pfSense.localdomain]/root: easyrule pass wan tcp 192.158.51.57 192.168.51.110 443*

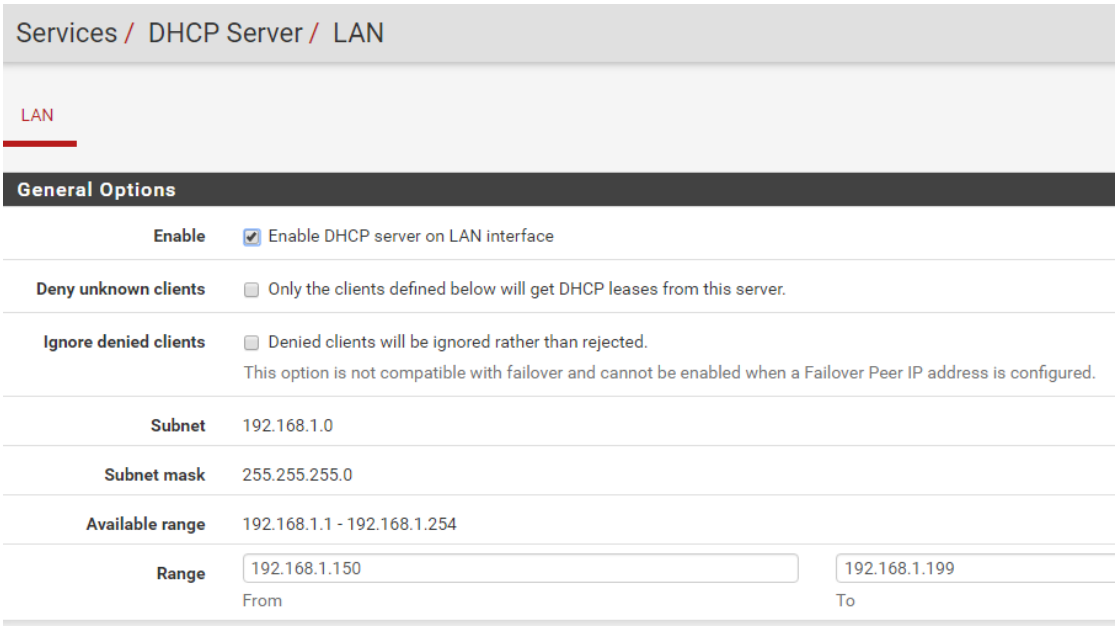
Komennon ensimmäinen osoite on sen koneen osoite, jolle haluttiin yhteys sallia ja jälkimmäinen on palomuurin WAN-rajapinnan osoite. Viimeinen parametri 443 vaaditaan, koska yhteys otetaan selaimen avulla salatulla *HTTPS*-protokollalla. Nyt voitiin asettaa selaimeen osoite *https://192.168.51.110/*, jonka jälkeen aukeaa kirjautumisikkuna. Oletuksena käyttäjätunnus on *admin* ja salasana *pfsense*. Graafisen käyttöliittymän etusivu näyttää kuvion 29 mukaiselta.



Kuvio 29. PfSensen graafinen käyttöliittymä

6.2.2 DHCP- ja DNS-palvelut

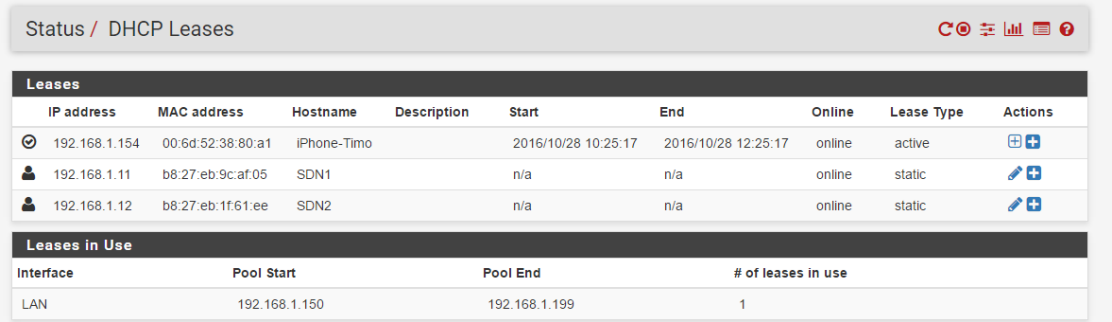
Seuraavaksi määriteltiin DHCP-asetuksiin osoitealue, josta asiakaslaitteet saavat dynaamisesti IP-osoitteen liityttäessä verkkoon. Nämä asetukset näkyvät kuviossa 30.



Kuvio 30. PfSensen DHCP-pool

On hyvä käytäntö, että tukiasemien ja muiden verkkolaitteiden hallintaosoitteet pysyvät aina samana. Koska tukiasemien hallintayhteyden rajapinta (*Br1)* on asetettu

hakemaan IP-osoitteensa DHCP-palvelimelta, tulee PfSenseen tehdä asetus, että tukiasemat saavat aina saman osoitteen DHCP-palvelimelta. Nämä osoitteet eivät kuulu samaan osoitealueeseen, kuin mistä asiakaslaitteet saavat osoitteensa. Tämä tehdään niin, että tukiasemien MAC-osoitteet asetetaan aina vastaanottamaan tietty IP-osoite. Asetus on esitelty kuviossa 31.



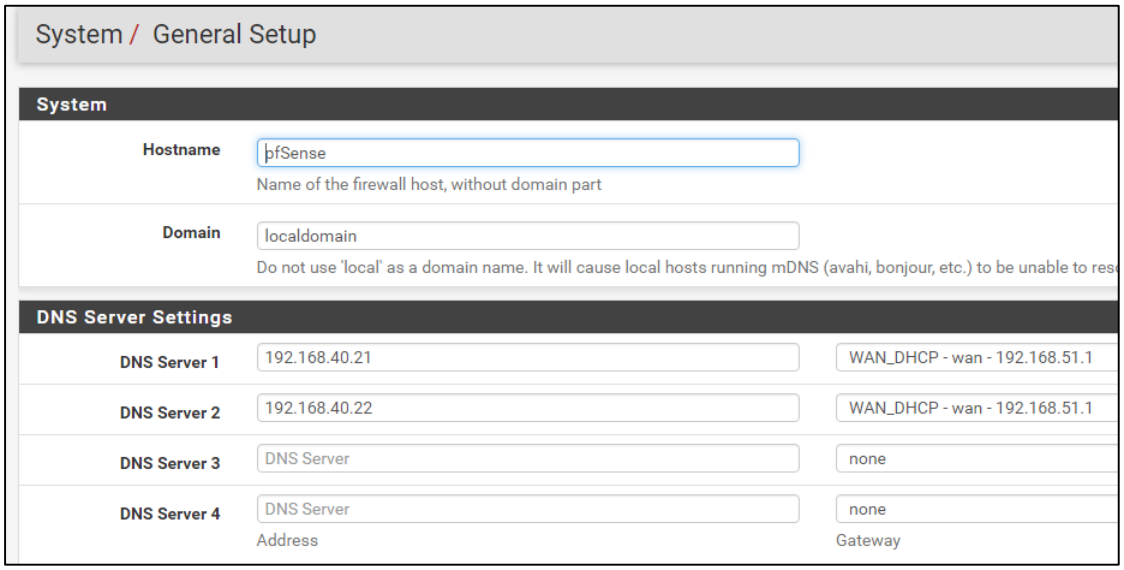
Kuvio 31. DHCP-palvelimen staattiset osoitteet

Kuten kuviosta näkyy, on DHCP-palvelimelta yksi asiakaslaite (*iPhone-Timo)* lainannut osoitteen dynaamisesti alueelta *192.168.1.150-.199*. Tukiasemien IP-osoitelainat sen sijaan ovat staattisia ja sidottu MAC-osoitteeseen. Tukiasemien hallintaosoitteet ovat seuraavanlaiset:

Taulukko 1. Tukiasemien staattiset hallintaosoitteet

Sijainti	**SSID**	**IP-osoite**
Kerros 1.	SDN1	*192.168.1.11*
Kerros 2.	SDN2	*192.168.1.12*
Kerros 3.	SDN3	*192.168.1.13*
Kerros 4.	SDN4	*192.168.1.14*

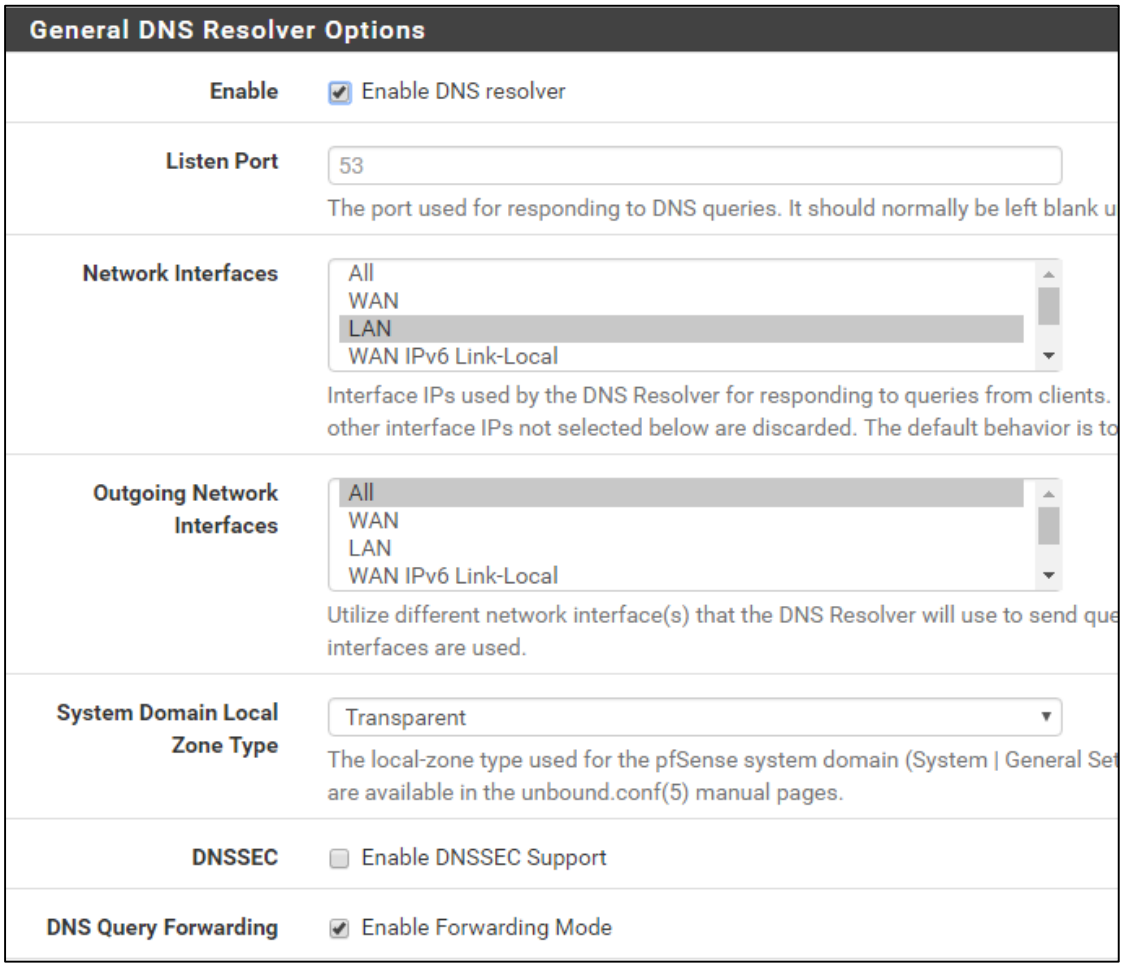
Jotta Internet-yhteys on mahdollinen LAN-verkosta, on DNS-palvelut määriteltävä niin, että WLAN-verkon asiakaslaitteet pääsevät käyttämään LabraNetin DNS-palvelimia *192.168.40.21* ja *192.168.40.22*. Tämä aloitetaan lisäämällä ko. palvelimen palomuurin yleisiin asetuksiin (*General Setup)*. Tämä on kuviossa 32.



Kuvio 32. DNS-palvelimet General Setup-asetuksissa

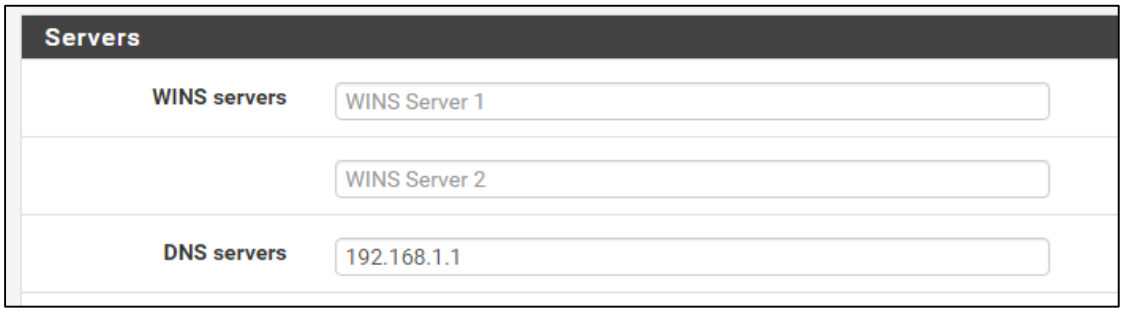
Samassa voidaan asettaa rajapinta, josta ko. palvelin löytyy, joka tässä tapauksessa oli LabraNetin reitittimen rajapinta palomuurin WAN-puolella. Tässä täytyi huomioida, että tämä määrittää vasta palomuurin DNS-palvelimet yleisesti. Tässä vaiheessa LAN-verkon asiakaslaitteet eivät tiedä vielä mitään näistä nimipalvelimista.

Luotiin seuraavaksi palomuurille ns. *Unbound DNS-server*, joka on palomuurilla toteutettu DNS-resolveri, jolta LAN-verkon asiakaslaitteet kysyvät vastauksia nimikyselyihin. Tämän luonti onnistui välilehdeltä *Services > DNS Resolver*. Tämä kuviossa 33.



Kuvio 33. DNS Resolver-asetuksen konfigurointi

Kuten kuviosta 33 nähdään, täytyi ensimmäiseksi asettaa palvelu voimaan kohdassa *Enable*. Tämän jälkeen valitaan rajapinnat, josta asiakaslaitteiden DNS-kyselyitä otetaan vastaan ja johon vastauksia palautetaan. Tähän tulee vaihtoehdot *LAN* ja *Localhost*. Oleellista oli myös asettaa voimaan vaihtoehto *Enable* kohtaan *DNS Query Forwarding*, sillä jos palomuurilla sijaitseva DNS Resolver ei tiedä vastausta, kykenee se välittämään kyselyn eteenpäin toisille DNS-palvelimille, jotka tässä tapauksessa olivat *General Setupissa* asetetut Labranetin DNS-palvelimet. Lopuksi täytyi DHCP-palveluun vielä määrittää asiakaslaitteiden rajapinta DNS-kyselyille. Tämä on kuvissa 34.

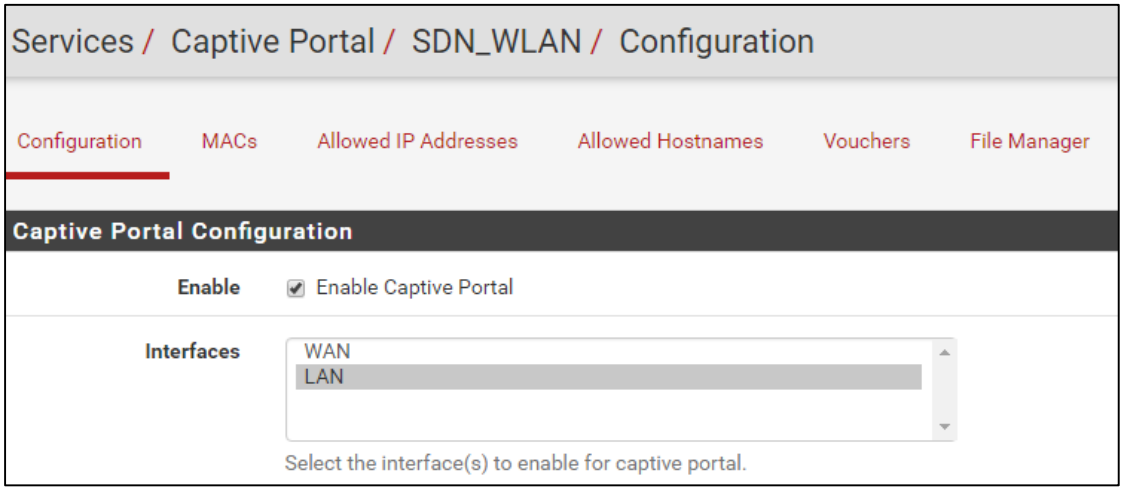


Kuvio 34. DNS-palvelimen määritys DHCP-palvelussa

Kuten kuviosta nähdään, on asiakaslaitteiden DNS-palvelimeksi määritetty PfSensen LAN-rajapinnan osoite *192.168.1.1*. Tämä siitä syystä, että PfSense hostaa DNS Resolver-palvelua, joka osaa lokaalisti vastata DNS-kyselyyn, jos se vastauksen tietää. Jos ei tiedä, se välittää (*DNS Query Forwarding)* kyselyn eteenpäin LabraNetin DNS-palvelimille. Mikäli LabraNetin palvelimetkaan eivät tiedä vastausta, välittävät ne kyselyn eteenpäin niin kauan, kunnes vastaus saadaan. Eli tärkeintä on ymmärtää, että DNS Resolverilla luotiin koko nimipalveluketjuun vain ylimääräinen solmu lisää.

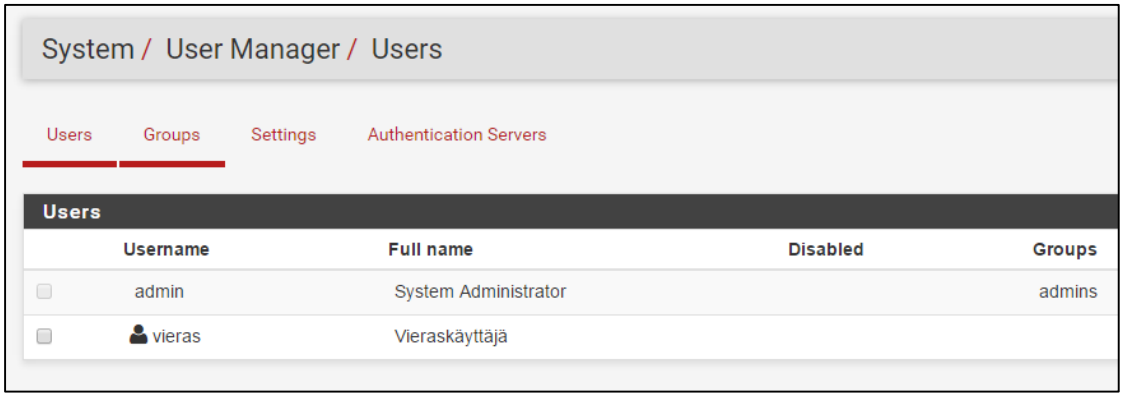
6.2.3 Captive Portal

Koska PfSensessä on sisäänrakennettu *Captive Portal*, ei tarvitse asentaa erillistä http-palvelinta verkkoon. Captive Portalin asennus voitiin aloittaa PfSensen välilehdeltä *Services > Captive Portal*, jonne luotiin uusi *Zone* nimellä SDN_WLAN. Tämä Zone tuli asettaa voimaan painamalla täppä ominaisuuteen *Enable* ja tämän jälkeen määrittää rajapinta, jolle palvelu halutaan käyttöön. Nämä ovat kuviossa 35.



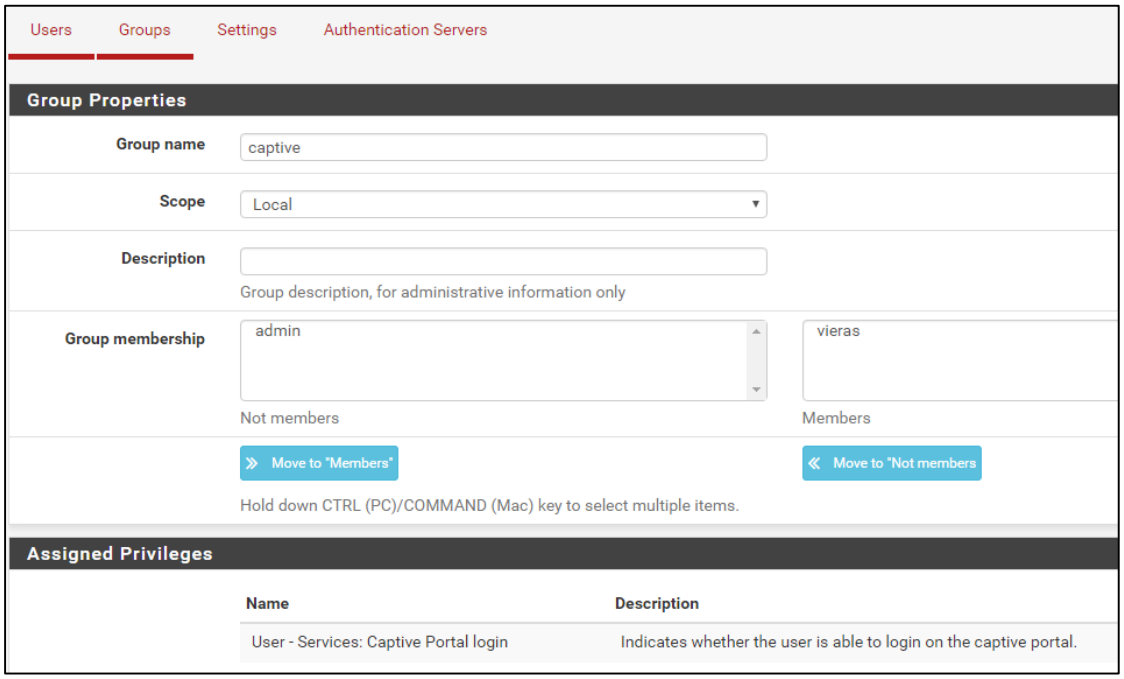
Kuvio 35. Captive Portalin käyttöönotto

Samalla konfiguraatiosivulla on monipuolisesti erilaisia vaihtoehtoja palveluun mm. autentikoinnin, sivusto-ohjauksien ja kaistankäytön suhteen, mutta tässä työssä jätettiin suurin osa oletusarvoille. Koska verkko oli toistaiseksi avoin, otettiin käyttöön yksinkertainen autentikointimetodi välilehdellä *Services > Captive Portal > SDN_WLAN > Authentication > Local User Manager/Vouchers*. Tällöin autentikointiin voitiin käyttää paikallisesti luotuja käyttäjiä. Vaihtoehtoisesti olisi ollut mahdollista myös käyttää ulkoista RADIUS-palvelinta. Käyttäjä luotiin palomuurin User Manager-välilehdellä, joka näkyy kuviossa 36.



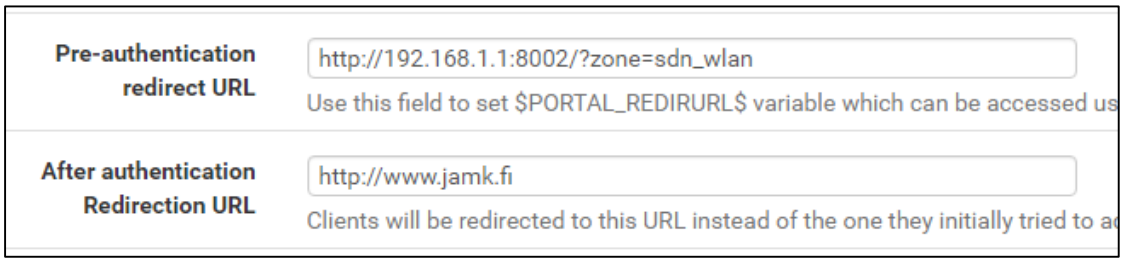
Kuvio 36. Local User Manager-näkymä

Tässä vaiheessa oli oleellista muistaa valita Captive Portalin asetuksista niin, että samalla käyttäjätunnuksella saa ottaa usean yhteyden, sillä muutoin jokaiselle käyttäjälle olisi joutunut tekemään käyttäjän erikseen. Käyttäjän lisäämisen jälkeen User Managerissa oli luotava uusi ryhmä, jonka itse nimesin nimellä *captive*. Tälle ryhmälle tuli asettaa käyttöoikeus kohdasta *Assigned Privileges* palveluun *Captive Portal Login* ja lisätä luotu käyttäjä *vieras* tähän ryhmään. Nämä on kuviossa 37.



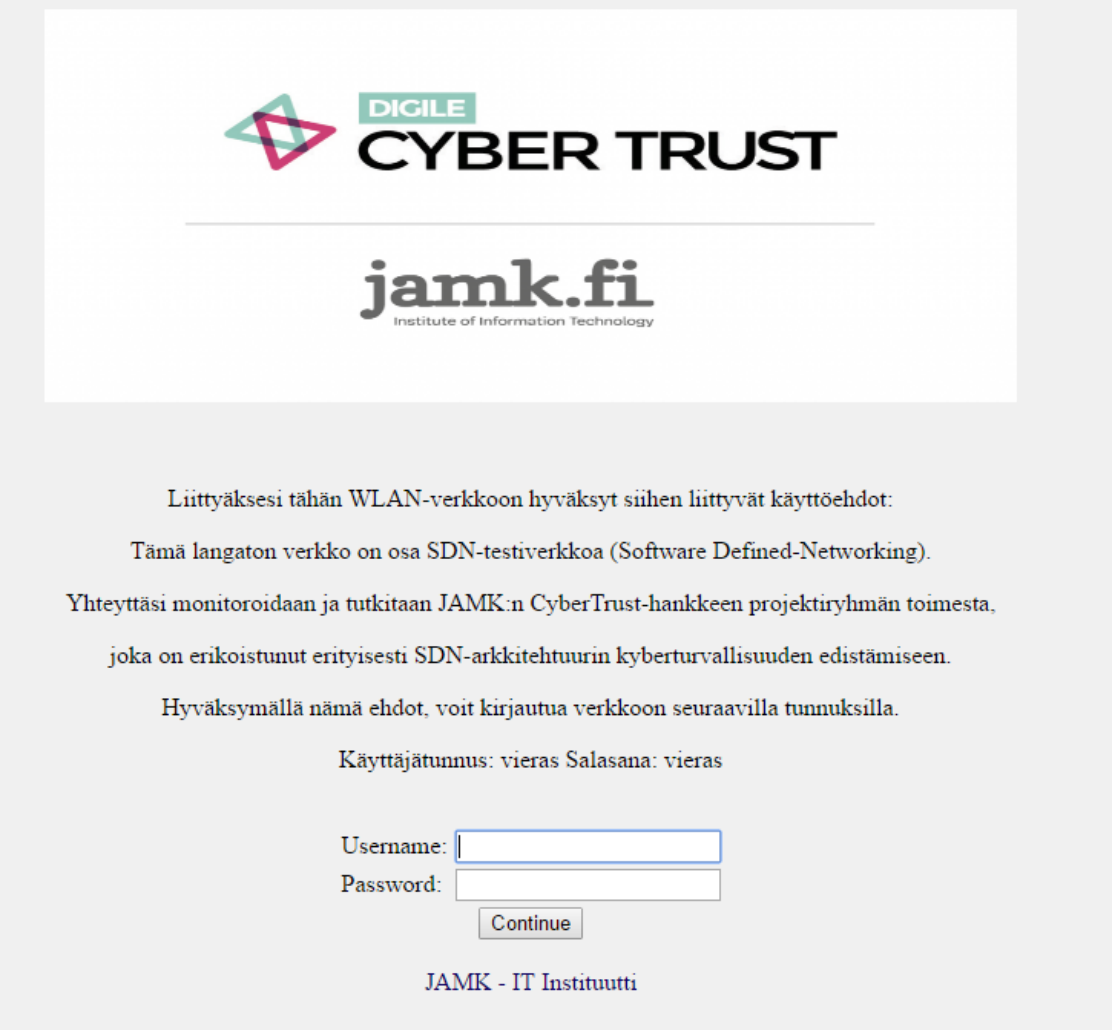
Kuvio 37. Captive Portal -ryhmän luonti ja käyttöoikeuden lisäys

Seuraavaksi tuli määrittää Captive Portal -profiilin asetuksiin, että mille sivulle käyttäjät ohjataan, kun yhteyttä yritetään muodostaa. Lisäksi voidaan määrittää, että mille verkkosivulle käyttäjä ohjataan, mikäli hän autentikoituu onnistuneesti ja yhdistyy verkkoon. Nämä on kuviossa 38.



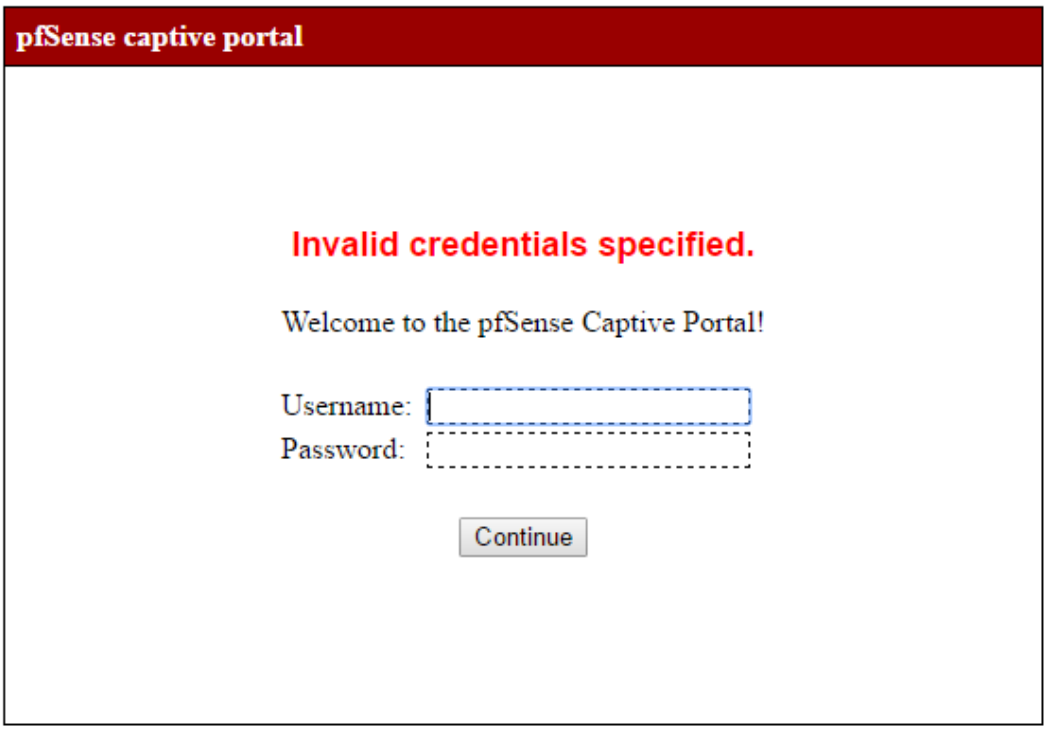
Kuvio 38. Captive Portalin ohjaussivut

Kuvion 36 osoite http://192.168.1.1:8002/?zone=sdn_wlan on verkkosivu, jota PfSense isännöi. Tämän sivun lähdekoodi tuli luoda erikseen. Tämän jälkeen latasin palomuurille *Portal.html*-tiedoston, joka luotiin portaalisivuksi käyttäjille, kun he liittyvät verkkoon. Tämän portaalisivun lähdekoodi on esitetty kokonaisuudessaan liitteessä 5. Kuva portaalisivusta, joka käyttäjille tulee aueta, näkyy kuviossa 39.



Kuvio 39. Captive Portal landing page

Sivu on siis ns. *landing page*, jolle käyttäjät tulee pakottaa, jotta he voivat lukea käyttöehdot liittyessään verkkoon. Kun käyttöehdot on hyväksytty, voi käyttäjä kirjautua paikallisilla tunnuksilla verkkoon, jonka jälkeen ohjataan käyttäjä sivulle *http://www.jamk.fi* ja hän voi aloittaa normaalin liikennöinnin ulkoverkkoon. Virheellisistä tunnuksista käyttäjälle täytyy tulla kuvion 40 mukainen ilmoitus.



Kuvio 40. Captive Portalin virheellinen kirjautuminen

6.3 ONOS-kontrolleri

ONOS-kontrolleria varten luotiin Virtualboxilla uusi virtuaalikone Ubuntu-käyttöjärjestelmällä. Huomioitavaa oli, että koneen verkkoadapteri oli asetettava siltaavaksi (*Bridged Adapter)* ja liittää se *TP-LINK Gigabit Ethernet* USB-adapteriin, jolloin se on nimenomaan LAN-puolelta yhteydessä PfSenseen. Todellisuudessa ONOS-kontrolleri tulee liittää verkkokaapelilla samaan OpenFlow-kytkimeen, mihin tukiasematkin kaapeloidaan, kun verkko tulevaisuudessa tulee tuotantoon.

Kontrollerin asennus voitiin tässä työssä tehdä yhdellä valmiilla skriptillä. Skripti on JAMK:n CyberTrust-projektin tuotos ja se asentaa kaikki tarvittavat pienohjelmat ja asetukset (OverFlowJAMK 2016.) Koko skripti seuraavanlainen:

```
apt-get update && \
echo debconf shared/accepted-oracle-license-v1-1 select true | debconf-set-
selections && \
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main" |
tee /etc/apt/sources.list.d/webupd8team-java.list && \
echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main"
| tee -a /etc/apt/sources.list.d/webupd8team-java.list && \
```

```
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886
&& \
apt-get update && \
apt-get install -y oracle-java8-installer oracle-java8-set-default && \
apt-get clean && apt-get purge && \
wget http://downloads.onosproject.org/release/onos-1.5.1.tar.gz && \
tar -xf onos-1.5.1.tar.gz && \
mv onos-1.5.1 onos
export HOME=/root
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export ONOS_ROOT=/root/onos
export KARAF_VERSION=3.0.5
export KARAF_ROOT=/root/onos/apache-karaf-3.0.5
export KARAF_LOG=/root/onos/apache-karaf-3.0.5/data/log/karaf.log
export PATH=$PATH:$KARAF_ROOT/bin
export ONOS_IP=localhost
```

Skriptillä asennettu ONOS on versiota 1.5.1. Kun ONOS on asennettu, voidaan sen palvelut käynnistää seuraavalla komennolla. Huomioitava ensin vaihtaa kansioon, jossa käynnistyksen skripti sijaitsee.

root@ONOS:~# cd /root/onos/bin

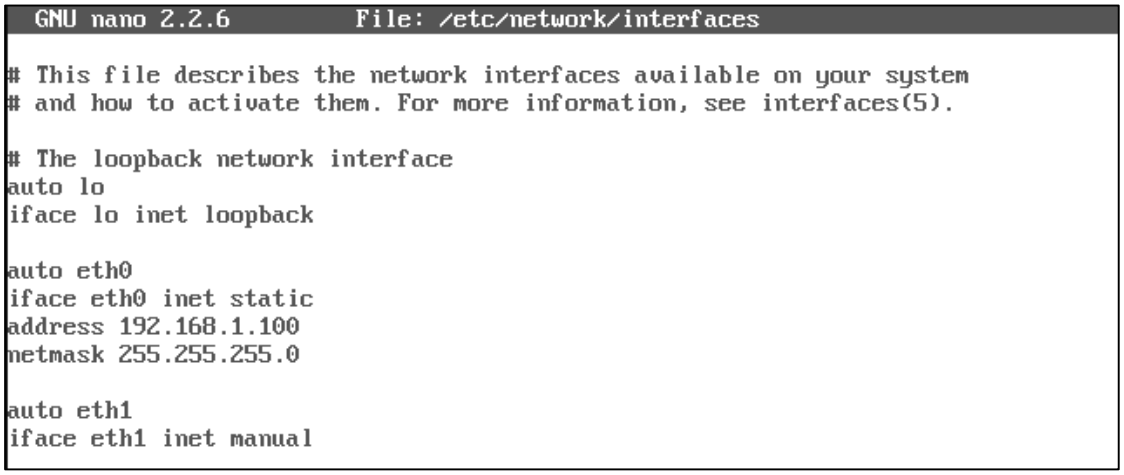
root@ONOS:~/onos/bin ./onos-service

Kuviossa 41 on esitelty ONOS:n komentorivipohjainen käyttöliittymä, joka tulee ruutuun, kun palvelun saa onnistuneesti käynnistettyä.

```
root@onos:~# cd /root/onos/bin
root@onos:~/onos/bin# ls
onos          onos-config        onos-jpenable    onos-service  onos-user-key
onos-client   onos-form-cluster  onos-secure-ssh  onos-ssh
root@onos:~/onos/bin# ./onos-service
Welcome to Open Network Operating System (ONOS)!
    ____  _  ______  ____
   / __ \/ |/ / __ \/ __/
  / /_/ /    / /_/ /\ \
  \____/_/|_/\____/___/

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos>
```

Kuvio 41. ONOS:n CLI-käyttöliittymä

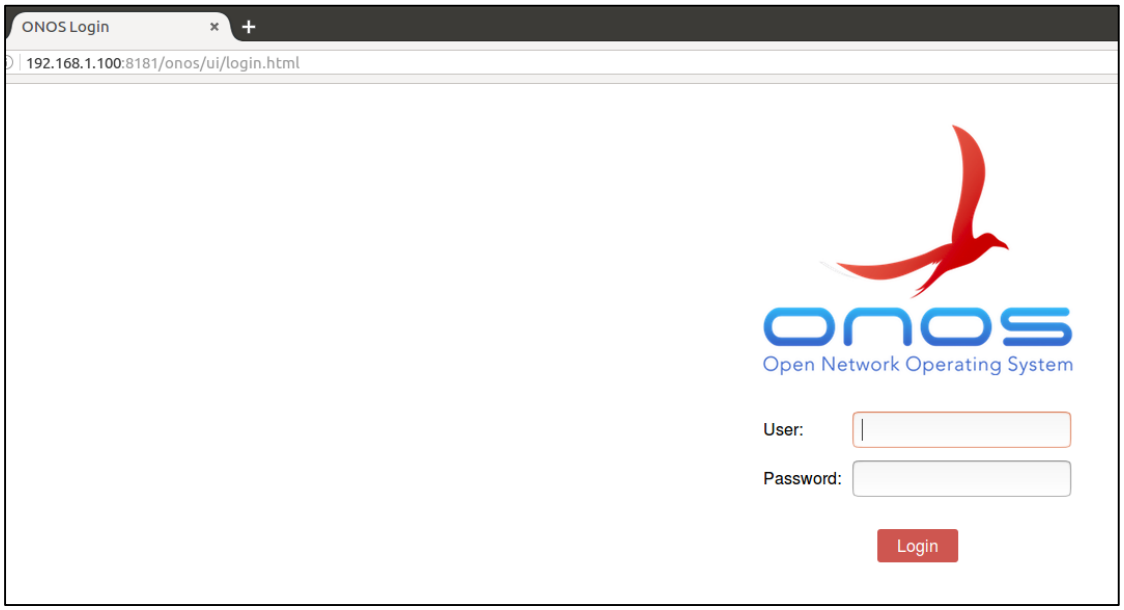
Koska WLAN-tukiasemien OVS:n virtuaalikytkimiin asetettu staattinen TCP-yhteys SDN-kontrollerille, on myös luonnollisesti itse SDN-kontrollerin IP-osoite oltava määritelty staattisesti samaan aliverkkoon. ONOS:n staattinen osoite määritelty polussa */etc/network/interfaces*. Tämä on havainnollistettu kuviossa 42.



Kuvio 42. ONOS:n interfaces-tiedosto

ONOS-kontrolleria on mahdollista myös hallita selainpohjaisella graafisella käyttöliittymällä. Kone, jolta hallintayhteys muodostetaan, täytyy olla samassa verkossa kontrollerin kanssa. Graafiseen käyttöliittymään pääsee kirjautumaan selaimella seuraavalla osoitteella: *http://192.168.51.100:8181/onos/ui/login.html*

Oletuksena käyttäjätunnus on *karaf* ja salasana *karaf*. ONOS:n graafisen käyttöliittymän käytöstä ja sen avulla verkkoelementtien keskitetystä hallinnasta on esittely laajemmin luvussa 7.2. Kuviossa 43 graafisen käyttöliittymän kirjautumisikkuna.



Kuvio 43. ONOS GUI:n kirjautumisikkuna

7 Toiminnan todennus ja analysointi

7.1 Yhteyden muodostus ja todennus

Seuraavaksi voitiin testata yhteyden muodostamista verkkoon. Kun PfSense ja ONOS-kontrolleri olivat käynnissä ja tukiasemat kaapeloitu kytkimeen kiinni, joka niin ikään on yhteydessä PfSenseen, voitiin käynnistää tukiasema, joka tässä todennuksen esimerkissä oli ensimmäisen kerroksen tukiasema. Kun palvelut ja asetukset tukiasemalla olivat nousseet ylös, tuli *SDN1*-verkko näkyviin matkapuhelimella ja se voitiin valita. Tämä on esitetty kuviossa 44.

Kuvio 44. Halutun WiFi-verkon valinta

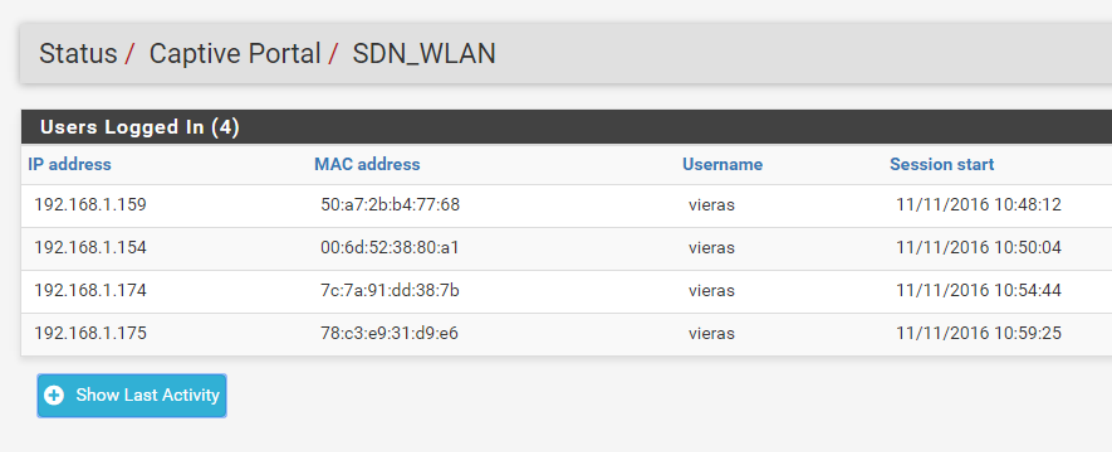
Kun haluttu *SDN1*-verkko valittiin asiakaslaitteella, pakotettiin käyttäjä välittömästi aloitussivulle, joka oli määritelty *Captive Portal* -asetuksissa (kts. kuvio 39). Tämä *landing page* on esitetty kuviossa 45.

Kuvio 45. Autentikointi SDN-verkkoon

Tämä aloitussivu skaalautuu paremmin esim. kannettavilla tietokoneilla koko ruudulle, mutta matkapuhelimella tarvittaessa voi zoomata kuvaa isommaksi. Kun tähän asettaa oikeat tunnukset, jotka käyttöehdoissa on esitetty, pääsee käyttäjä yhdistymään verkkoon ja hänet ohjataan automaattisesti sivulle *www.jamk.fi*, joka oli asetettu Captive Portalin asetuksissa. Tämä on esitetty kuviossa 46.

Kuvio 46. Autentikoinnin jälkeinen ohjaus

Tässä vaiheessa käyttäjä on yhdistynyt verkkoon ja on vapaa aloittamaan normaalin liikennöinnin ulkoverkkoon. Tässä vaiheessa verkkoon yhdistynyt laite myös näkyy PfSensen tiedoissa välilehdellä *Status > Captive Portal > SDN_WLAN*. Tämä on esitetty kuviossa 47.

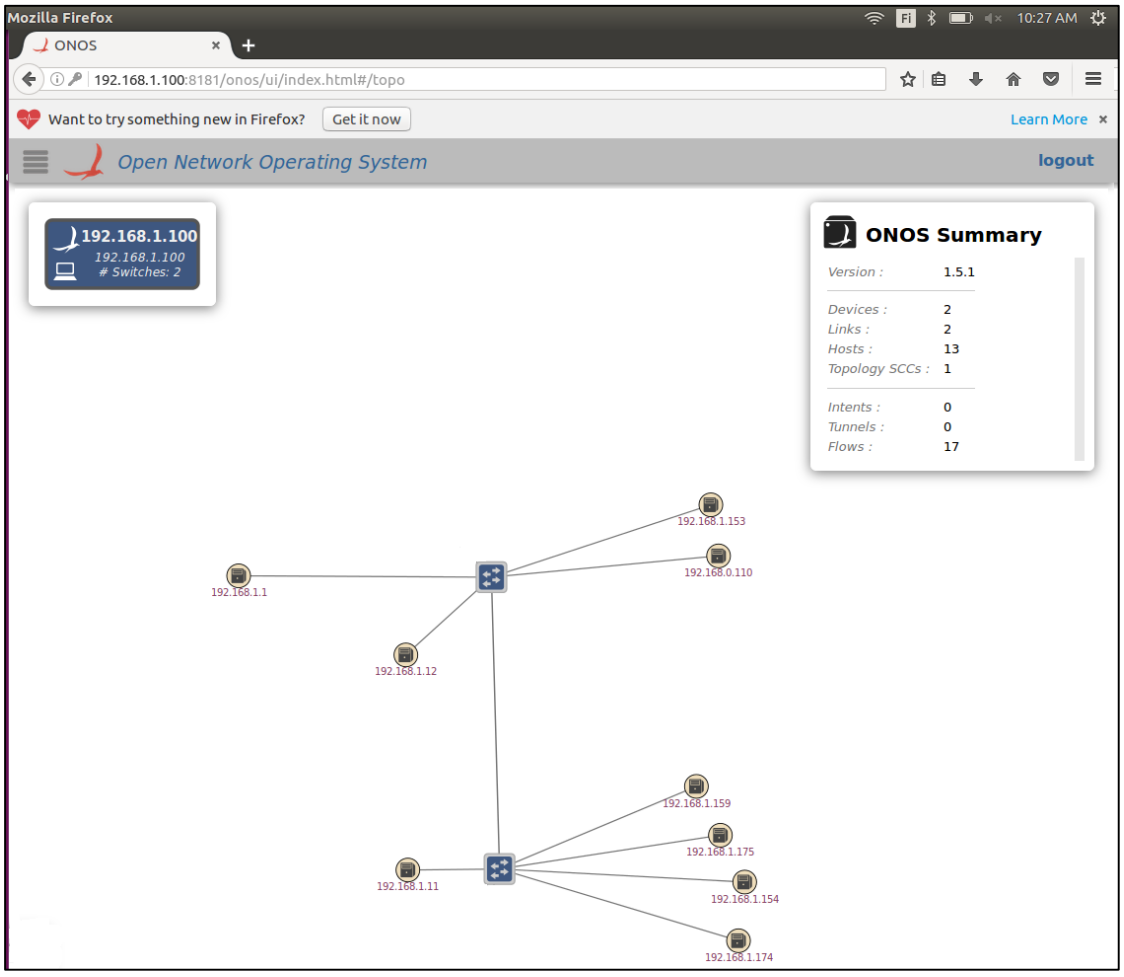


Kuvio 47. SDN-verkkoon yhdistyneet laitteet

Listassa näkyy asiakaslaitteen saama IP-osoite, MAC-osoite ja istunnon aloitusaika. Kaikissa yhteyksissä käyttäjä on *vieras*, koska oli asetettu niin, että samoilla käyttäjätunnuksilla saa ottaa useamman yhteyden. Tietyn asiakaslaitteen yhteyden katkaisu myös onnistuu manuaalisesti tältä samalta sivulta.

7.2 Hallinta kontrollerilta

Kun useilla asiakaslaitteilla liityttiin testiksi pariin eri langattomaan verkkoon (SDN1 ja SDN2), voitiin kirjautua ONOS-kontrollerin graafiseen käyttöliittymään ja todentaa sieltä verkkoon liittyneet laitteet. Tämä on havainnollistettu kuviossa 48.



Kuvio 48. ONOS-kontrollerilla hallittavat asiakaslaitteet

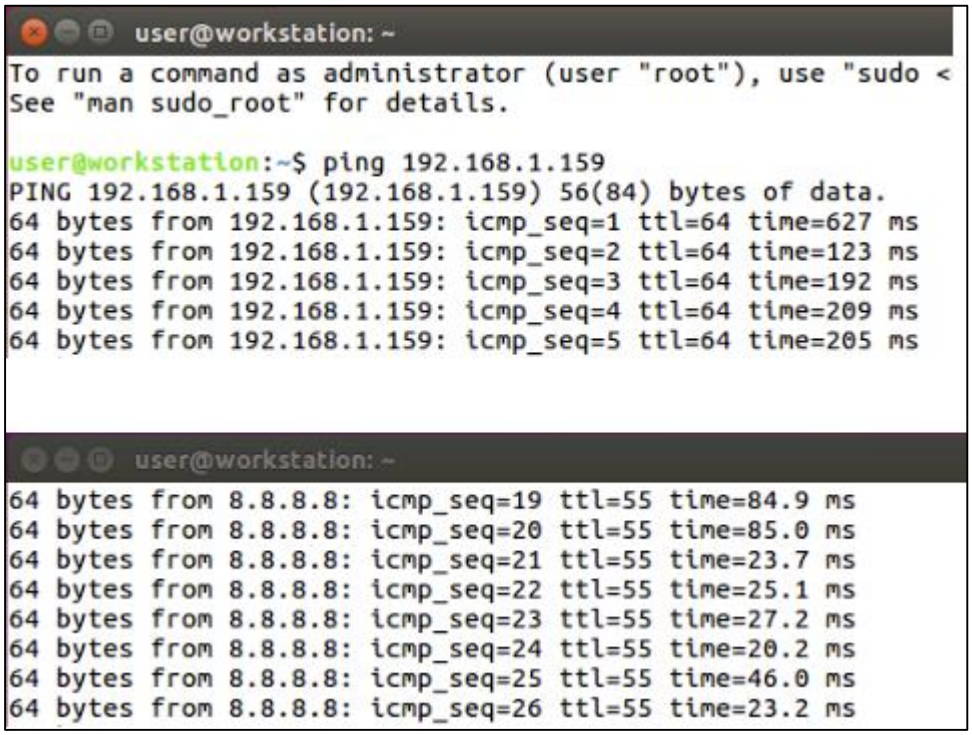
Kuviossa näkyvät siniset kytkimet kuvaavat fyysisiä tukiasemia. Molempien tukiasemien vasemmalla puolella näkyy niiden MAC-osoitteisiin sidotut hallintaosoitteet, jotka ovat alemmalle SDN1-tukiasemalle *192.168.1.11* ja ylemmälle SDN2-

tukiasemalle *192.168.1.12* (kts. taulukko 1). Oikealle sen sijaan on raahattu erilleen tukiaseman kautta verkkoon yhteydessä olevat langattomat asiakaslaitteet.

Tukiasemien yhteys PfSenselle, joka näkyy instanssina äärivasemmalla osoitteella *192.168.1.1,* näkyy virheellisesti sen takia, että tukiasemat ovat testiverkon aikana olleet tavalliseen Ciscon L2-tason hallittavaan kytkimeen kaapeloitu ja ovat tämän kautta verkkokaapelilla yhteydessä PfSenselle. Koska Ciscon L2-tason kytkimessä ei ole OpenFlow-protokollalle tukea, se ei myöskään näy ONOS-kontrollerilla. Todellisuudessa tuo Ciscon kytkin korvattaisiin kytkimellä, jossa on tuki OpenFlow:lle (kts. kuvio 12) ja näin ollen topologia ONOS-käyttöliittymässäkin näyttäisi oikealta.

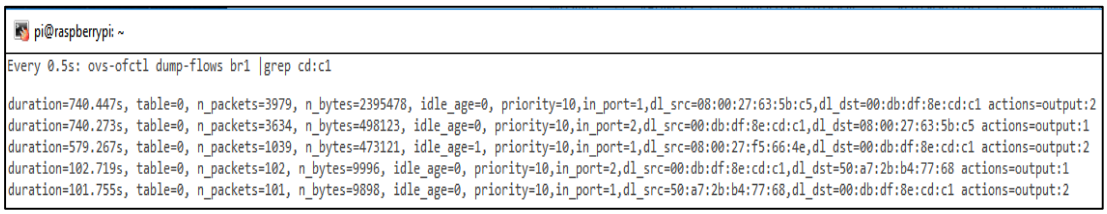
Seuraavaksi voidaan todentaa, miten tällä graafisella käyttöliittymällä voidaan tehdä helposti manipulointia tukiasemien vuotauluihin. ONOS-kontrollerilla ei voi tehdä vuotaulumerkintöjä, jossa paketteja pudotetaan pois (*action=drop*), vaan sillä voidaan vaikuttaa siihen, mistä OpenFlow-tukiasemien tai -kytkinten porteista liikennettä välitetään ulos. Muilla SDN-kontrollereilla pakettien pudotuskin onnistuu, joka on myös tuotantoverkkoihin vaadittava ominaisuus.

Otetaan esimerkki edelleen kuviosta 48, josta käytetään osapuolta. Tukiasemaan SDN2 on yhteydessä kannettava tietokone osoitteella *192.168.1.111*, jolla MAC-osoite *00:DB:DF:8E.CD:C1*. Testin toinen asiakaslaite on älypuhelin ja yhteydessä tukiasemaan SDN1 osoitteella *192.168.1.159*, jolla MAC-osoite *50:A7:2B:B4:77:68*. SDN2-verkon kannettavalta tietokoneelta laitetaan jatkuva Ping-komento juoksemaan sekä SDN1-verkon älypuhelimelle, että ulkoverkkoon Googlen DNS-palvelimelle. Tämä kuviossa 49.



Kuvio 49. SDN2-asiakkaan Ping-komento ulkoverkkoon ja SDN1-verkon asiakkaalle

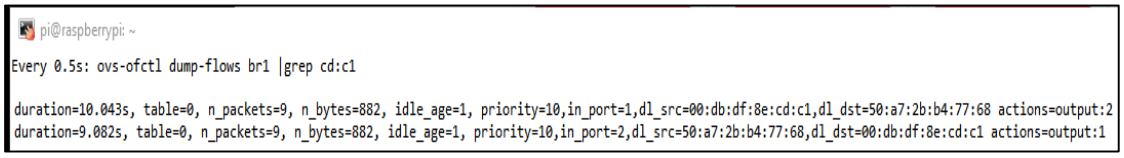
ONOS-kontrolleri asettaa tässä vaiheessa OpenFlow-laitteiden eli tässä tapauksessa tukiasemien vuotauluihin reaktiivisen merkinnän näille liikenteille. Kuviossa 50 näkyy SDN2-tukiaseman vuotaulumerkinnät. Huomiona, että komennossa on haettu MAC-osoitteen lopulla vain näiden asiakaslaitteiden liikennettä koskevia merkintöjä yksinkertaisuuden vuoksi.



Kuvio 50. SDN2-tukiaseman reaktiiviset vuotaulumerkinnät

Kuten kuviosta voidaan todeta, on SDN2-tukiasemalla vuotaulussa merkinnät koskien sekä ulkoverkon että sisäverkon Ping-komentoa. Kun lähteenä eli *src* on asiakaslaite itse, on liikenne välitettävä ulospäin portista 1 eli *actions=output:1*. Kun taas liikenteen kohteena eli *dst* on asiakaslaite itse, on kytkimen välitettävä se sisään portista 2

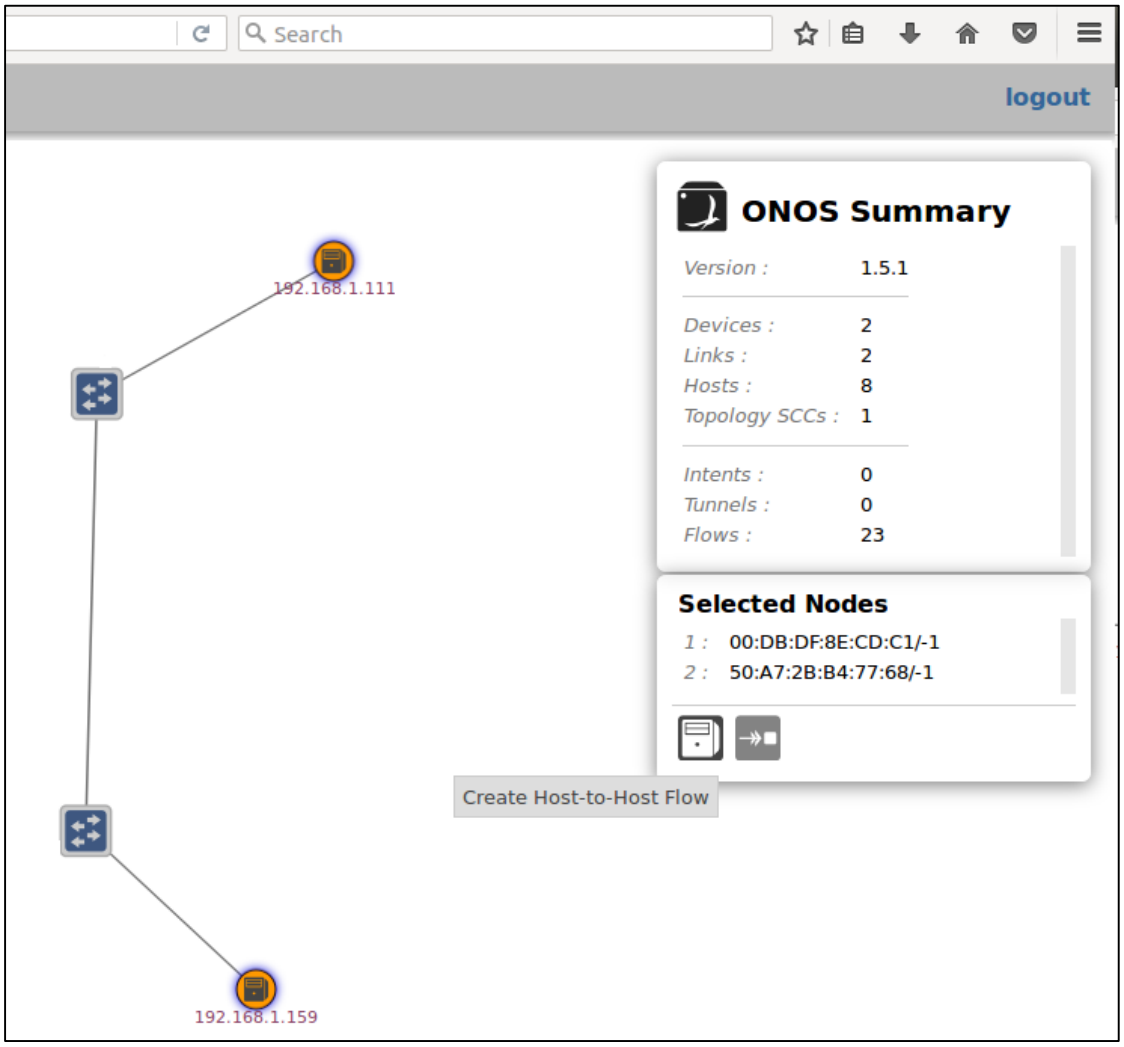
eli *actions=output:2.* Tärkeänä huomiona tässä vaiheessa, että reaktiivisten eli ns. dynaamisesti tehtyjen vuotaulumerkintöjen prioriteetti on 10. Viides vuotaulumerkintä kuviossa tulee SSH-yhteydestä tukiasemalle ja siitä ei tarvitse tässä välittää. Sama komento suoritettiin seuraavaksi tukiasemalla SDN1 ja tämän tuloste on kuviossa 51.



Kuvio 51. SDN1-tukiaseman reaktiiviset vuotaulumerkinnät

Kuten kuviosta voidaan todeta, on siinä vähemmän vuomerkintöjä, kuin aikaisemmassa, joka johtuu siitä, että SDN1-tukiaseman ei tarvitse käsitellä ollenkaan kannettavan tukiaseman Ping-komentoa Googlen DNS-palvelimelle, koska se on jo välitetty SDN2-tukiasemalta PfSenselle ja sitä kautta kohteeseensa. Vuotaulussa näkyy ainoastaan kannettavan tietokoneen Ping-komento älypuhelimelle, koska se on ko. tukiasemaan yhteydessä. Tämä voidaan todeta sillä, että kun kohteena on älypuhelimen MAC-osoite, välitetään liikenne sisään portista 2. Vastaavasti kun lähteenä on älypuhelin, on tukiaseman tehtävänä välittää se ulospäin portista 1.

Tässä vaiheessa todennusta kaikki merkinnät vuotauluihin on tehty ONOS-kontrollerin toimesta reaktiivisesti eli se reagoi automaattisesti. Seuraavaksi luodaan vuotauluihin merkintöjä proaktiivisesti eli ne luodaan itse käsin ennakoivasti. Tässä voidaan käyttää hyväksi ONOS:n graafista käyttöliittymää. Käyttöliittymässä se onnistuu niin, että valitaan ne asiakaslaitteet, joiden välille merkintä halutaan luoda ja valitaan sen toiminta. Tämä vaihe on esitetty kuviossa 52.



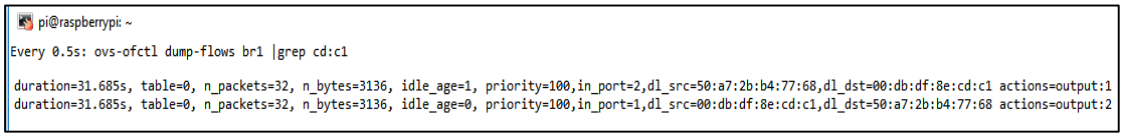
Kuvio 52. Vuotaulumerkinnän luonti graafisessa käyttöliittymässä

Kuten kuviosta käy ilmi, on siinä valittu samat asiakaslaitteet, kuin aikaisemmin ja oikean hiiren klikkauksella pääsee valitsemaan toiminnot, jotka niin ikään näkyvät kuviossa. Luodaan proaktiivinen vuo valitsemalla toiminto *Create Host-to-Host Flow.* Seuraavaksi tarkastetaan uudestaan molempien tukiasemien vuotaulut ja todetaan toiminta. Ensimmäiseksi tukiasema SDN2 ja tämän tuloste kuviossa 53.

```
pi@raspberrypi: ~
Every 0.5s: ovs-ofctl dump-flows br1 |grep cd:c1

duration=1118.526s, table=0, n_packets=7664, n_bytes=5730086, idle_age=1, priority=10,in_port=1,dl_src=08:00:27:63:5b:c5,dl_dst=00:db:df:8e:cd:c1 actions=output:2
duration=1118.352s, table=0, n_packets=5954, n_bytes=804498, idle_age=1, priority=10,in_port=2,dl_src=00:db:df:8e:cd:c1,dl_dst=08:00:27:63:5b:c5 actions=output:1
duration=957.346s, table=0, n_packets=2148, n_bytes=961633, idle_age=1, priority=10,in_port=1,dl_src=08:00:27:f5:66:4e,dl_dst=00:db:df:8e:cd:c1 actions=output:2
duration=94.995s, table=0, n_packets=95, n_bytes=9310, idle_age=0, priority=100,in_port=2,dl_src=00:db:df:8e:cd:c1,dl_dst=50:a7:2b:b4:77:68 actions=output:1
duration=94.995s, table=0, n_packets=96, n_bytes=9408, idle_age=0, priority=100,in_port=1,dl_src=50:a7:2b:b4:77:68,dl_dst=00:db:df:8e:cd:c1 actions=output:2
```

Kuvio 53. SDN2-tukiaseman proaktiiviset vuotaulumerkinnät

Kuviosta voidaan tulkita, että vuotaulumerkinnät ovat samat ulkoverkkoon päin, koska niihin ei tehty muutoksia, mutta kannettavan tietokoneen liikenne älypuhelimelle nousi prioriteettiin 100. Alla on esitetty todennus vuotaulusta SDN1-tukiasemalta.



Kuvio 54. SDN1-tukiaseman proaktiiviset vuotaulumerkinnät

Jälleen vähemmän merkintöjä, koska SDN1 käsittelee testissä vain asiakaslaitteiden välistä liikennettä, mutta proaktiivisesti luodun vuon jälkeen myös tässä tukiasemassa liikenteen prioriteetti nousi arvoon 100.

Näistä voidaan todeta se, että manuaalisesti luotu proaktiivinen vuo yliajaa reaktiivisesti luodut vuomerkinnät. Jos ONOS-kontrollerilla voisi tehdä pakettien pudotusta, ei liikenne asiakaslaitteiden välillä toimisi tässä vaiheessa, koska itse luodun ennakoivan vuon prioriteetti 100 yliajaa dynaamisesti luodut vuot prioriteetilla 10, vaikka niiden toiminta olisi mikä tahansa. Mutta koska työn aiheena ei ollut kontrolleripuolen toiminta, niin näinkin yksinkertainen todennus riittää todistamaan sen, että asiakaslaitteiden liittäminen langattomasti SDN-verkkoon niin, että niitä voidaan hallita keskitetysti OpenFlow-protokollaa hyväksikäyttäen, toimii mainiosti.

8 Pohdinta

Työn tavoitteista saavutettiin suurin osa. Langaton verkkoratkaisu saatiin toteutettua niin, että langattomia tukiasemia voitiin hallita kontrollerilta. Lisäksi verkkoon liittyminen sisältää kevyen autentikoinnin ja käyttäjille käy ilmi verkon käyttämisen ehdot. Ainoa tavoite, jota ei tullut työssä lähestyttyä käytännössä lainkaan, oli sFlow-työkalu, jolla voidaan keskitetysti monitoroida verkon liikennettä.

Työn aikana ilmeni useita erilaisia ongelmia, joista ensimmäinen koski suoraan koko työn aloittamista. Koko Software-Defined Networking oli tekniikkana itselleni täysin uusi, joten käytin alussa merkittävän osan aikaa lukien ja kirjoittaen teoriaa aiheesta. Tämän jälkeen sain hyvät ohjeet siihen, miten aloittaa käytännössä työhön tutustumaan ja pienten testiympäristöjen toimiessa ymmärsi yhä enemmän SDN-tekniikkaa, kun käytäntöä pääsi jo hieman opitulla teorialla perustelemaan ja toteamaan.

Seuraavat ongelmat liittyivätkin ihan vianetsintään ja Linux-käyttöjärjestelmään. Vaikka Linux-käyttöjärjestelmien peruskäyttö sujui, niin varsinkin syvällisempi vianetsintä oli välillä työlästä, joskin onneksi virheistä ja ongelmista löytyy verkosta paljon materiaalia. Mutta suuremmalla Linux-kokemuksella olisin säästänyt tuntitasolla selvästi aikaa. Lisäksi useita pienempiä hidasteita, kuten huono HTML-osaaminen, topologian jatkuva muuttuminen ja ajoittain huonommat mahdollisuudet fyysisesti päästä koululle tekemään käytännön osuutta välillä vaikuttivat työn etenemiseen. Muutoin teknisesti ongelmia ei alun jälkeen ollut vaivaksi asti, vaan pienten testiympäristöjen toimiessa tiesin, että mitä tulee tehdä, mutta luonnollisesti tuli selvittää, että miten.

Selviä jatkokehityksiä työlle kuitenkin on useita. Esimerkiksi tukiasemista (Raspberry Pi) voisi helposti saada pienellä vaivalla vikasietoisempia ja nopeampia käynnistämään palvelut. Jokainen tukiasema myös toistaiseksi mainostaa omaa SSID:tä eli voisi olla hyvä idea tutkia, onnistuuko tukiasemille tehdä ns. roaming-ominaisuutta, jolla kaikki tukiasemat mainostaisivat samaa SSID:tä ja asiakaslaitteet

osaisivat itse aloittaa liikennöinnin lähempänä olevaan tukiasemaan, kun signaali alkaa liikkuessa olemaan liian heikko edelliseen.

Lisäksi langaton siirtotie päätelaitteiden ja tukiasemien välillä on täysin salaamatonta ja autentikointi toteutettu ainoastaan mallilla käyttäjätunnus-salasana. Koska CyberTrust-hanke nimensä mukaisesti pureutuu myös kyberturvallisuuteen, on aivan selvää, että aidossa tuotantoverkossa liikenne tulisi jollain menetelmällä salata ja autentikaation tulisi vahvempi (esim. ulkoinen RADIUS-palvelin).

Opinnäytetyön aihe oli mielestäni mielenkiintoinen ja koin sen aidosti hyödylliseksi. Jos aikaa olisi jäänyt enemmän, olisin toteuttanut varmastikin osan mainitsemistani kehitysideoista, mutta ennen kaikkea olisin opetellut käyttämään itse SDN-kontrolleria paremmin, vaikka se ei työn päämäärä ollutkaan. Olen lopputulokseen suhteellisen tyytyväinen, sillä se tieto SDN-arkkitehtuurista, millä työhön lähdin, ei täytä edes määritystä perusteet.

Lähteet

About the IETF, N.d. Tietoa Internet -standardien julkaisijasta. Viitattu 11.9.2016. https://www.ietf.org/about/

Bridging Network Connections. 2016. Esittely Bridge-utils virtuaalikytkimestä, asennuksesta ja konfiguroimisesta. Viitattu 21.10.2016. https://wiki.debian.org/BridgeNetworkConnections

DIMECC Cyber Trust Program: We Return the Trust to the Digital World, n.d. DIMECC -sivulla esittely Cyber Trust -hankkeesta. Viitattu 7.9.2016. http://cybertrust.fi/

DIMECC Securing platforms and networks, n.d. DIMECC -sivulla esittely Work package 3 -sisällöstä. Viitattu 7.9.2016. http://cybertrust.fi/research-themes/securing-platforms-and-networks/

Flowgrammable, 2016. OpenFlow Switch. Tutkijoista ja insinööreistä koostuvan liiton verkkojulkaisu OpenFlow-kytkimestä ja sen rakenteesta. Viitattu 19.9.2016. http://flowgrammable.org/sdn/openflow/#tab_switch

Ghosh, A. 2014. Why Software Defined Data Center is Emerging in IT? Verkkojulkaisu SDN :stä datakeskuksissa. Viitattu 10.9.2016. https://thecustomizewindows.com/2014/02/why-software-defined-data-center-is-emerging-in-it/

Haleplidis, E., Pentikousis, K., Denazis, S., Hadi Salim, J. 2015. RFC 7426: Software-Defined Networking (SDN): Layers and Architecture Terminology. IETF :n julkaisema standardi SDN :stä. Viitattu 8.9.2016. https://tools.ietf.org/html/rfc7426#section-1

Holm, S. 2014. WLAN-järjestelmän käyttöönotto. Opinnäytetyö Tampereen Ammattikorkeakoululta. Viitattu 21.11.2016. https://www.theseus.fi/bitstream/handle/10024/72798/Simo_Holm.pdf?sequence=1

Introducing pfSense, 2016. Pfsense-organisaation sivusto ohjelman asennuksesta, ominaisuuksista ja dokumenteista. Viitattu 21.10.2016. https://doc.pfsense.org/index.php/Introducing_pfSense

LitePoint 2013. IEEE 802.11ac: What Does it Mean for Test? Standardista luotu dokumentti. Viitattu 21.11.2016. http://litepoint.com/whitepaper/80211ac_Whitepaper.pdf

Malinen, J. 2013. Hostapd: IEEE 802.11 AP. Tietoa Hostapd-daemonista. Viitattu 21.10.2016. http://w1.fi/hostapd/

McKeown Group, 2010. OpenFlow Overview. Stanford -yliopiston tutkimusryhmän verkkosivuilla julkaistu artikkeli OpenFlow :sta. Viitattu 19.9.2016. http://yuba.stanford.edu/cs244wiki/index.php/Overview

ONF: Software-Defined Networking, 2012. The New Norm for Networks. Verkkojulkaisu SDN -verkon tarpeesta ja hyödyistä. Viitattu 8.9.2016.

Open Networking Foundation, 2014. SDN Architecture Overview version 1.1. ONF :n versio SDN -arkkitehtuurista. Viitattu 8.9.2016.

https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf

ONF: What is ONF? n.d. Open Networking Foundation -organisaation sivuille linkitetty PDF -tiedosto. Viitattu 14.9.2016. https://www.opennetworking.org/images/stories/downloads/about/onf-what-why-2016.pdf

ON.LAB, 2014. Introducing ONOS - a SDN network operating system for Service Providers. Viitattu 14.11.2016. http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf

Open vSwitch, n.d. OVS -kytkimen kehittäjäorganisaation verkkosivut. Viitattu 23.9.2016. http://openvswitch.org/

OverFlowJAMK, 2016. JAMK:n CyberTrust-projektin valmistama skripti ONOS-kontrollerin asennukseen. Viitattu 31.10.2016. https://github.com/OverFlowJAMK/General/wiki/How-to-install-ONOS-version-1.5.1

Pakzad, F., Portmann, M., Tan Lum, W., Indulska, J., 2014. Efficient Topology Discovery in Software Defined Networks. Kappale PDF -tiedostona IEEE :n sivuilla julkaistusta e-kirjasta. Viitattu 16.9.2016.

RGCE – Kybertoimintaympäristö, n.d. JYVSECTEC -verkosivujen esittely RGCE -verkkoon ja sen ominaisuuksiin. Viitattu 7.9.2016. http://jyvsectec.fi/fi/kyberymparisto/

Tietoa JAMKista, N.d. Ammattikorkeakoulun verkkosivuilta tietoa organisaatiosta. Viitattu 6.9.2016. https://www.jamk.fi/fi/Tietoa-JAMKista/

Verry, T. 2012. What is the Raspberry Pi? Blogijulkaisu Raspberry Pi -tietokoneesta. Viitattu 17.9.2016. http://www.extremetech.com/computing/124317-what-is-raspberry-pi-2

Vähä-Touru, T. 2007. Langattoman lähiverkon toteutus. Opinnäytetyö Tampereen Ammattikorkeakoululta. Viitattu 21.11.2016. https://publications.theseus.fi/bitstream/handle/10024/10134/TMP.objres.1013.pdf?sequence=2

Welcome to Rasbian, n.d. Rasbian -käyttöjärjestelmän kehittäjien voittoa tavoittelemattoman organisaation verkkosivujen esittely tuotteesta. Viitattu 23.9.2016. https://www.raspbian.org/

What are SDN Controllers? N.d. SDX Centralin verkkojulkaisu kontrollereista ja niiden historiasta. Viitattu 16.9.2016. https://www.sdxcentral.com/sdn/definitions/sdn-controllers/

What are SDN Northbound APIs? N.d. SDX Centralin verkkojulkaisu Northbound API -rajapinnan toiminnasta. Viitattu 17.9.2016. https://www.sdxcentral.com/sdn/definitions/north-bound-interfaces-api/

What are SDN Southbound APIs? N.d. SDX Centralin verkkojulkaisu Southbound API -rajapinnan toiminnasta. Viitattu 17.9.2016. https://www.sdxcentral.com/sdn/definitions/southbound-interface-api/

Liitteet

Liite 1. Raspberry Pi 3.0 :n tekniset tiedot

Raspberry Pi 3 - Model B Technical Specification

- Broadcom BCM2387 chipset
- **1.2GHz Quad-Core ARM Cortex-A53**
- **802.11 bgn Wireless LAN and Bluetooth 4.1** (Bluetooth Classic and LE)
- **1GB RAM**
- **64 Bit CPU**
- 4 x USB ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- 10/100 BaseT Ethernet socketbr
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source

Liite 2. Open vSwitch -kytkimen toiminnallisuudet

Open vSwitch supports the following features:

- Visibility into inter-VM communication via NetFlow, sFlow(R), IPFIX, SPAN, RSPAN, and GRE-tunneled mirrors
- LACP (IEEE 802.1AX-2008)
- Standard 802.1Q VLAN model with trunking
- Multicast snooping
- IETF Auto-Attach SPBM and rudimentary required LLDP support
- BFD and 802.1ag link monitoring
- STP (IEEE 802.1D-1998) and RSTP (IEEE 802.1D-2004)
- Fine-grained QoS control
- Support for HFSC qdisc
- Per VM interface traffic policing
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- OpenFlow protocol support (including many extensions for virtualization)
- IPv6 support
- Multiple tunneling protocols (GRE, VXLAN, STT, and Geneve, with IPsec support)
- Remote configuration protocol with C and Python bindings
- Kernel and user-space forwarding engine options
- Multi-table forwarding pipeline with flow-caching engine
- Forwarding layer abstraction to ease porting to new software and hardware platforms

Liite 3. Hostapd-daemonin ominaisuudet

Supported WPA/IEEE 802.11i/EAP/IEEE 802.1X features

- WPA-PSK ("WPA-Personal")
- WPA with EAP (with integrated EAP server or an external RADIUS backend authentication server)
- key management for CCMP, TKIP, WEP104, WEP40
- WPA and full IEEE 802.11i/RSN/WPA2
- RSN: PMKSA caching, pre-authentication
- IEEE 802.11r
- IEEE 802.11w
- RADIUS accounting
- RADIUS authentication server with EAP
- Wi-Fi Protected Setup (WPS)

Liite 4. PfSensen ominaisuudet

- Web interface (Uses SSL by default)
- Wireless support
- Stateful packet filtering
- NAT/PAT (including 1:1)
- DHCP client, PPPoE and PPTP support on the WAN interface
- IPsec VPN tunnels (IKE; with support for hardware crypto cards)
- OpenVPN for site-to-site or remote access setup
  - Remote Acesss OpenVPN also supports local, RADIUS, or LDAP backed authentication
- PPTP VPN (with RADIUS server support)
- Static routes
- DHCP server or relay
- Caching DNS forwarder
- DynDNS client
- SNMP agent
- Traffic shaper
- Firmware upgrade through the web browser or console
- Configuration backup/restore
- Host, network, and port aliases
- High-availability active/passive failover using CARP
- Captive Portal
- Multi-WAN Load Balancing (outbound)
- Server Load Balancing (inbound)
- NTP Server
- PPPoE Server (with RADIUS server support)
- Universal Plug-n-Play (UPnP, NAT-PMP)
- Wake on LAN

Liite 5. Captive Portal-sivun html-koodi

```
<html>
<head>
<title>JAMK:n CyberTrust-hankkeen Software Defined-Netowrking WLAN-verkko</title>
<style type="text/css">
.style1 {
		font-family: "Adobe Garamond Pro";
		color: #0B0061;
}
.style2 {
		font-size: medium;
}
.style5 {
		font-family: "Adobe Garamond Pro";
		color: #0B0061;
		font-size: medium;
}
</style>
</head>
<body style="background-color: #F0F0F0">
<center>
<h2 class="style1">
<img src="http://jyvsectec.fi/wp-content/uploads/2016/05/cybertrust_jyvsectec_blog-1024x598.png"
width="634" height="256"></h2>
<p class="style1"> </p>
<span class="style2">Liittyäksesi tähän WLAN-verkkoon hyväksyt siihen liittyvät käyttöehdot:
<p>Tämä langaton verkko on osa SDN-testiverkkoa (Software Defined-Networking).</p>
 <p>Yhteyttäsi monitoroidaan ja tutkitaan JAMK:n CyberTrust-hankkeen projektiryhmän toimesta,</p>
 <p>joka on erikoistunut erityisesti SDN-arkkitehtuurin kyberturvallisuuden edistämiseen.</p>
 <p>Hyväksymällä nämä ehdot, voit kirjautua verkkoon seuraavilla tunnuksilla.</p>
 <p>Käyttäjätunnus: vieras Salasana: vieras</p>
<br>
</span>
<form method="post" action="$PORTAL_ACTION$"">
```

```
<input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
<table>
  <tr><td class="style2">Username:</td><td><input name="auth_user" type="text"></td></tr>
  <tr><td class="style2">Password:</td><td><input name="auth_pass" type="password"></td></tr>
  <tr>
   <td colspan="2">
                    <center><input name="accept" type="submit" value="Continue"></center>
   </td>
  </tr>
</table>
</form>

<p class="style5"> JAMK - IT Instituutti<br>
<br>
</p>
<!--

 Up, Up, Down, Down, Left, Right, Left, Right, B, A, START!


-->

</html>
```